UNIVERSITÀ DEGLI STUDI DI PISA

FACOLTÀ DI SCIENZE MATEMATICHE FISICHE E NATURALI

CORSO DI LAUREA SPECIALISTICA IN INFORMATICA

TESI DI LAUREA

FastType: Predizione di Parola basata su Modelli Statistici in un Ambiente di Scrittura Assistita

CANDIDATA

Nedjma Deha

RELATORI

Prof. Paolo Mancarella

Dott. Carlo Aliprandi

CONTRORELATORE

Dott. Vincenzo Gervasi

Anno Accademico 2005/2006

Ai miei Genitori,

a mia sorella e mio fratello, Ibtissem e Hicham,

ed a Enrico

Per il loro amore e incorraggiamento.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Il ruolo dell'informatica nella vita delle persone diversamente abili assume una sempre maggiore rilevanza. I computer stanno aprendo nuove strade all'integrazione sociale e professionale di dette persone, che fino a qualche anno fa, erano confinate nello spazio della loro condizione psicomotoria. Per le persone che presentano disabilità comunicative, le tecnologie dell'informazione si stanno infatti rivelando sempre più indispensabili, sia come strumento di studio e di lavoro, che come mezzo per costruire e mantenere relazioni interpersonali, aspetto quest'ultimo imprescindibile per una soddisfacente qualità di vita. In molti casi l'utilizzo del computer è l'unico mezzo per raggiungere determinati obiettivi: pensiamo ad esempio a tutti quei casi in cui non sia possibile utilizzare la tradizionale scrittura con carta e penna perché un deficit motorio impedisce l'esecuzione di questo compito. Tuttavia spesso la menomazione rende difficile se non impossibile l'utilizzo

del computer: mouse, tastiera e output visivo attraverso il monitor, per chi presenta disabilità motorie o sensoriali divengono un ostacolo insormontabile per l'accesso al computer. Frequentemente le barriere comunicative possono essere superate grazie ad ausili informatici, appositamente studiati, per facilitare l'uso delle attitudini neuromotorie personali e rendere di conseguenza il computer, e tutti i vantaggi che esso comporta, accessibile a chiunque. La ricerca in questo campo applicativo porta alla creazione di software innovativi, volti ad elevare il lavoro e la vita relazionale di persone con importanti capacità intellettuali che altrimenti rimarrebbero inespresse. La mancanza di tali supporti tecnologici si traduce spesso in un vero e proprio isolamento intellettuale che può e deve essere superato.

Nell'ambito dei software creati per favorire l'efficienza della comunicazione di queste persone, si possono trovare diversi tipi di sistemi, che si diversificano in base al tipo di disabilità. Tastiere speciali (espanse, ridotte, con scudo, virtuali), schermo tattile, mouse a trackball o a joystick, comando vocale, sintesi vocale, programmi a scansione, sono nati per sostituire i sistemi standard di input (mouse e tastiera) e output (monitor), offrendo la possibilità di adattare lo strumento computer anche a chi presenta difficoltà, utilizzando e valorizzando le capacità residue, anche nei casi in cui siano limitate all'uso di un solo senso, di un solo arto, ecc. L'utilizzo degli ausili informatici, oltre a favorire gli apprendimenti, ha delle positive ricadute su aree come l'autostima, la memoria, l'attenzione, la socializzazione

e di conseguenza l'autonomia.

Per le persone con disabilità motorie, che non sono in grado di usare i dispositivi di inserimento del testo, risulta molto utile avere a disposizione uno strumento che faciliti la scrittura e che faccia loro risparmiare tempo: si pensi che un dattilografo esperto può superare le 400 battute al minuto[1] su una tastiera tradizionale, mentre la velocità di battitura di un disabile motorio risulta essere molto inferiore, intorno ai 100 caratteri al minuto[2]. Una possibile alternativa all'inserimento di testo potrebbe essere quella del riconoscimento vocale. Ma questa soluzione risulta essere non applicabile a tutte le disabilità, in particolare in presenza di persone con difficoltà nella comunicazione verbale e nel caso in cui l'utente si trovi in un luogo dove non si può far rumore (per esempio uno studente in biblioteca).

Il lavoro di questa tesi si colloca proprio nell'ambito della facilitazione della comunicazione scritta per le persone disabili.

1.1 Tecnologia Assistiva

Con il termine "Assistive Technology" o "Tecnologia Assistiva" viene definito quell'insieme di prodotti o servizi a base tecnologica concepiti per facilitare la vita alle persone portatrici di deficit motori, sensoriali o cognitivi.

[1] Si pensi che i campioni di scrittura nelle gare mondiali all'Intersteno, raggiungono una velocità di scrittura tra le 800 e le 900 battute al minuto (`http://www.intersteno.it/`).

[2] In base ad una sperimentazione effettuata dall'USID (Unità di Servizi per il sostegno e l'Integrazione degli studenti Disabili) dell'Università di Pisa su un campione di studenti universitari disabili.

Il concetto ispiratore di tale insieme di strumenti è sempre quello di fornire "assistenza" alle capacità comunicative, siano esse indirizzate alla terapia, alla riabilitazione, all'apprendimento o alla compensazione delle limitazioni psico-fisiche.

Le persone disabili, molto spesso, hanno bisogno di strumenti per sopperire ad una mancanza, ad un deficit. Ad esempio, una persona non vedente non avrà grossi problemi nell'uso della tastiera: tutte le dattilografe veloci e precise da sempre hanno usato le macchine da scrivere guardando solo il foglio da copiare. Il problema per una persona non vedente è di sapere quale sarà la risposta del calcolatore ad un comando immesso. Allo stesso modo, una persona con problemi motori, ad esempio con difficoltà nell'uso delle mani, pur essendo in grado di controllare lo schermo, avrà difficoltà nell'uso degli strumenti di immissione, tastiera o mouse. Se queste persone hanno ausili adeguati a superare il loro deficit, ecco che usare un computer diventa facile; anzi, attenua le barriere di esclusione e di differenziazione che si formano, loro malgrado, intorno ai diversamente abili.

Lo straordinario sviluppo delle tecnologie dell'informazione e della comunicazione ha fatto emergere una nuova categoria di ausili informatici, che hanno aperto nuove possibilità di sviluppo, per il miglioramento delle condizioni di vita delle persone diversamente abili. Alcuni esempi di tali tecnologie sono la barra Braille, che permette ai non vedenti di utilizzare il tatto come strumento per la comprensione del testo, oppure i programmi

di sintesi vocale, che permettono di usare l'udito per la lettura dei testi. Allo stesso modo i programmi di ASR (Automatic Speech Recognition, ossia riconoscimento del parlato) e di trascrizione automatica permettono ai non udenti di usare la vista per la lettura delle registrazioni vocali. Altri ausili informatici rivolti in particolare ai disabili motori sono tastiere e mouse speciali, uso di comandi vocali, sintesi vocale, programmi a scansione con l'obiettivo di sostituire i sistemi di input (mouse e tastiera) e output (monitor) standard.

Un'altra tipologia di ausili particolarmente utili nei casi in cui la limitazione allo scrivere sia dovuta a difficoltà motorie o del linguaggio, sono sistemi di accelerazione dell'input nella composizione di testi. Un efficace sistema di predizione del testo, che consenta di ridurre al minimo il numero di battute (e quindi i movimenti necessari) e di scrivere parole corrette ortograficamente o, nel caso di disabili cognitivi, che riduca gli errori di scrittura, è in questi casi importante.

Ogni ausilio deve avere come punto di arrivo la persona, le sue esigenze, le sue difficoltà, e deve adattarsi alle sue capacità.

La tecnologia assistiva tramite questi ausili contribuisce in maniera determinante all'autonomia delle persone con deficit motori, sensoriali e cognitivi, favorendone la partecipazione sociale, l'integrazione lavorativa e l'indipendenza economica, l'accesso alla cultura e alle attività ricreative e quindi amplia le capacità di pensare, di informarsi, di esprimersi, accelerando ed

accrescendo le possibilità di comunicazione e di controllo.

1.2 Comunicazione Aumentativa e Alternativa

La Comunicazione Aumentativa e Alternativa (CAA)[3] è "ogni comunicazione che aumenta o sostituisce il linguaggio verbale", ed è "un'area della pratica clinica che cerca di compensare la disabilità temporanea o permanente di individui con bisogni comunicativi complessi, attraverso l'uso di componenti comunicativi speciali o standard" [2]. Si tratta quindi di un intervento e non di una tecnica, e come tale può utilizzare tante "tecniche" diverse.

L'aggettivo "Aumentativa" (traduzione dal termine inglese Augmentative) indica come le modalità di comunicazione utilizzate siano tese non a sostituire, ma ad accrescere la comunicazione naturale, utilizzando tutte le competenze dell'individuo ed includendo le vocalizzazioni o il linguaggio verbale residuo, i gesti, i segni, la comunicazione con ausili e tecnologie avanzate.

Il termine "Alternativa" viene usato sempre meno perché presuppone di sostituire le modalità comunicative esistenti.

La comunicazione aumentativa è quindi il termine usato per descrivere l'insieme di conoscenze, di strategie e di tecnologie che è possibile attivare per facilitare la comunicazione delle persone che presentano menomazioni della parola, della funzione linguistica e della scrittura. Si tratta di un approc-

[3]Il termine Comunicazione Aumentativa Alternativa deriva dall'inglese Augmentative Alternative Communication, (AAC).

cio che tende a creare opportunità di reale comunicazione anche attraverso tecniche, strategie e tecnologie atte a coinvolgere la persona che utilizza la CAA e il suo ambiente di vita.

1.3 La Predizione di Parola

La predizione di parola è una tecnica spesso usata nei sistemi di comunicazione aumentativa e alternativa con l'obiettivo di velocizzare l'inserimento, la qualità e la correttezza del testo. Un predittore linguistico è uno strumento capace di individuare quali parole o completamento di parole hanno la maggiore probabilità di seguire un dato segmento di testo, ossia quale sarà la parola che l'utente vorrebbe scrivere.

Un programma che implementa un meccanismo di predizione di parole, comunemente chiamato predittore o word predictor, opera leggendo il carattere o la sequenza di caratteri inseriti dall'utente e genera una lista di parole che corrisponde ai possibili suggerimenti. L'utente può scegliere una parola tra i suggerimenti oppure continuare inserendo un nuovo carattere fino a quando la parola che desidera scrivere compare tra i suggerimenti. Il suggerimento selezionato dall'utente viene poi automaticamente inserito nel testo.

Per realizzare la predizione, un predittore ha bisogno di avere accesso ad un modello del linguaggio, ovvero ad una descrizione che cattura i pattern e le regole presenti nel linguaggio naturale, sfruttando competenze ed

informazioni di carattere statistico e morfosintattico.

Le informazioni relative al linguaggio su cui si basa la predizione, variano a seconda del sistema di predizione. Alcuni predittori infatti si basano esclusivamente sulla frequenza delle singole parole, non tenendo conto del contesto nel quale esse sono inserite, mentre altri sistemi affiancano alle informazioni di carattere statistico ulteriori informazioni sul linguaggio di carattere grammaticale e morfosintattico, consentendo così il mantenimento della concordanza tra parole.

1.4 Benefici della Predizione

La tipologia di utenti che traggono maggiore beneficio dall'utilizzo di un predittore è quella delle persone portatrici di disabilità motorie agli arti superiori e di conseguenza aventi difficoltà nel processo di scrittura con la tastiera.

Usando un word predictor, queste persone possono migliorare le proprie prestazioni, infatti la predizione della parola corretta risparmia all'utente l'inserimento di tutti i caratteri rimanenti, aumentando significativamente la velocità di inserimento del testo, riducendo lo sforzo necessario alla composizione delle singole parole e permettendo all'utente disabile di prolungare l'attività di scrittura.

Un'altra categoria di utenti che beneficiano in modo particolare della predizione di parole è costituita da persone che soffrono di disturbi del lin-

guaggio (come per esempio la dislessia), ossia difficoltà nel produrre frasi ortograficamente e grammaticalmente corrette o difficoltà nel trovare le parole. In questo caso l'uso di uno strumento di predizione assicura la corretta composizione del testo, che ortograficamente corretto, permette un risparmio in termini di numero di battute necessarie alla scrittura, comportando un minor sforzo per l'utente.

Se il predittore è dotato di informazioni riguardanti la grammatica della lingua, i suggerimenti offerti permetteranno inoltre di scrivere frasi grammaticalmente corrette.

Un altro campo di applicazione della predizione della parola è quello dell'apprendimento della lingua, in quanto i suggerimenti offerti dal predittore completano la capacità dell'utente di richiamare alla memoria la parola desiderata.

Recentemente le tecniche di predizione sono state prese in considerazione nell'ambito di nuovi domini applicativi. In particolare risultano essere utili per gli utenti di dispositivi mobili sprovvisti di appropriati supporti di inserimento testo, come telefoni cellulari, computer palmari e personal organizers. In questi dispositivi l'inserimento di testo spesso risulta fastidiosamente lento e inefficiente a causa del numero limitato di tasti e della mancanza di metodi alternativi veloci. L'utilizzo di un modulo di predizione, tuttavia, impone un maggior sforzo cognitivo dovuto al fatto che l'utente è costretto a spostare l'attenzione tra testo in fase di inserimento e suggerimenti offerti da

selezionare. In persone affette da disabilità cognitive ciò potrebbe ridurre i vantaggi offerti da un predittore. Per questo l'accurata scelta del numero di suggerimenti, del loro posizionamento e l'uso di tastiere virtuali su schermo permette di ridurre questo sforzo.

1.5 Il Progetto Precedente ed il Nuovo Predittore

L'attività svolta per questa tesi rientra nel progetto di sviluppo di un sistema di word prediction. Scopo di tale progetto è la realizzazione di un sistema efficace per la predizione di parola.

Il lavoro principale è stato quello di migliorare le capacità predittive del sistema precedente [14]. Per raggiungere questo obiettivo sono state necessarie delle modifiche importanti al prototipo esistente e l'aggiunta di nuove funzionalità.

Una parte fondamentale del lavoro è stata quella dell'adattamento e dell'organizzazione di risorse linguistiche esistenti: dizionari, linguaggio e grammatiche fornite dalla Società Synthema[4], oltre all'aggiunta di nuove risorse linguistiche.

Inoltre, è stato sviluppato un nuovo algoritmo di predizione. Il nuovo predittore chiamato "*FastType*" propone meccanismi di predizione sulla base di risorse linguistiche diverse. Ognuno di questi meccanismi di predi-

[4]**SYNTHEMA** nasce nel 1994 a Pisa dall'iniziativa di alcuni specialisti del Centro di Ricerca **IBM**, che costituiscono una società per la traduzione automatica, la localizzazione, Data et Text Mining.

zione determina un modello del linguaggio, che viene usato per ottenere una predizione parziale e che sfrutta un sottoinsieme delle risorse linguistiche disponibili. La predizione finale è data da una combinazione lineare pesata delle predizioni prodotte dai singoli meccanismi che effettuano la predizione sfruttando un modello del linguaggio basato su diverse risorse linguistiche.

D'altra parte, usando l'autoapprendimento, ovvero la capacità di integrare il lessico personale dell'utente, il sistema permette la costruzione di un dizionario personale, aumentando così il numero di suggerimenti appropriati.

1.6 Schema della Tesi

I capitoli sono strutturati secondo il seguente schema:

Nel Capitolo 2 viene presentata un'analisi dei modelli del linguaggio alla base dei sistemi predittivi attuali.

Nel Capitolo 3 sono descritte le risorse linguistiche usate per la realizzazione di questo sistema e quindi del modulo di predizione.

Nel Capitolo 4 viene presenta la descrizione e la realizzazione del progetto di questa tesi ed in particolare gli algoritmi usati per il predittore di parola e le innovazioni introdotte.

Il Capitolo 5 descrive gli esperimenti effettuati, le metriche di valutazione del sistema e i risultati ottenuti.

Il Capitolo 6 illustra lo stato attuale di sviluppo del sistema, riassume le conclusioni a cui si è giunti e fa il punto sui possibili sviluppi futuri.

Capitolo 2

Modelli del Linguaggio Naturale

2.1 Introduzione

In un sistema di predizione ricavare un modello del linguaggio naturale è un passo fondamentale perché il sistema possa svolgere le funzioni di predizione. Questo modello deve essere in grado di descrivere le caratteristiche e le regolarità (patterns) della lingua.

Esistono due tipi di modelli linguistici: modelli statistici e modelli knowledge-based (basati sulla conoscenza).

Un modello statistico è fondato sull'estrazione automatica di dati da vaste raccolte di testi detti corpora[1], mentre un modello knowledge-based si

[1] plurale di corpus. Il corpus è un insieme di testi su cui si fonda la descrizione grammaticale di una lingua.

affida alle generalizzazioni linguistiche definite, come le regole grammaticali.

Generalmente i sistemi di predizione di parola fanno uso di modelli statistici, ma alcuni di essi si avvalgono di informazioni predefinite sul linguaggio come quelle usate dai modelli knowledge-based. Altri predittori usano modelli ibridi cercando di ottenere un modello più rappresentativo e inerente al linguaggio naturale.

2.2 Modelli Statistici

I modelli statistici del linguaggio esprimono la probabilità a priori di sequenze di parole, asserendo che una sequenza di parole W appartiene al linguaggio con probabilità $\mathbb{P}(W)$. I modelli del linguaggio più comuni sono modelli statistici ad n-gram, normalmente bigram ($n = 2$) e trigram ($n = 3$). Un modello bigram esprime la probabilità che la parola w_2 sia preceduta dalla parola w_1, mentre un modello trigram descrive la probabilità che la parola w_3 sia preceduta dalla sequenza w_1 w_2. Si tratta di modelli usati inizialmente nel riconoscimento vocale, ma recentemente estesi in ambiti applicativi più tipici del trattamento del linguaggio scritto, per esempio per ricavare automaticamente ontologie[2] nel campo dell'estrazione di informazioni o per classificare le parole di un testo all'interno di categorie morfo-sintattiche.

Le probabilità dei modelli statistici del linguaggio vengono stimate a

[2]una **ontologia** è il tentativo di formulare uno schema concettuale esaustivo e rigoroso nell'ambito di un dato dominio; si tratta generalmente di una struttura dati gerarchica che contiene tutte le entità rilevanti, le relazioni esistenti fra di esse, le regole, gli assiomi, ed i vincoli specifici del dominio.

partire da insiemi di testi e frasi, sia generali della lingua che specifici del contesto applicativo.

2.2.1 Word Unigram

Il più semplice modello del linguaggio sviluppato agli inizi degli anni '80, usato nei sistemi di predizione, si basa sulla probabilità delle singole parole ed è chiamato modello unigram. Il modello unigram assegna ad ogni parola w una probabilità $\mathbb{P}(w)$ calcolata in base alla frequenza di occorrenza della parola, estratta da un dato corpus.

Sia $C(w)$ il numero di occorrenze della parola w nel corpus C di dimensione $|C|$ (numero totale delle parole nel corpus). Allora:

$$\mathbb{P}(w) = \frac{C(w)}{|C|}$$

Il modello assume che tutte le parole siano scelte indipendentemente l'una dall'altra, ovvero che la probabilità di una sequenza di parole $w_1, w_2, \ldots, w_n$ sia il prodotto delle probabilità di ciascuna parola. Formalmente:

$$\mathbb{P}(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} \mathbb{P}(w_i) = \mathbb{P}(w_1) \cdot \mathbb{P}(w_2) \cdot \ldots \cdot \mathbb{P}(w_n)$$

Supponiamo che l'utente abbia digitato una frase e che la sequenza di

parole inserita per ultima sia la seguente:

$$\ldots\ w_1\ w_2\ w_{prefix}$$

dove w_1 e w_2 sono le ultime due parole inserite completamente e w_{prefix} è il prefisso della parola corrente, ossia l'ultima parola che l'utente sta digitando.

Definiamo W come l'insieme di tutte le parole del lessico (corpus) che iniziano con il prefisso w_{prefix}. Il predittore che usa tale modello ordina W in base alla frequenza d'uso di tali parole.

Quindi per esempio una sequenza

"La il casa mamma"

sarà una sequenza altamente probabile poiché ognuna di queste parole, prese singolarmente, ha un'alta probabilità, in quanto il suo uso è molto frequente in un dato corpus. Questo modello non risulta interessante per la predizione di parola in quanto l'assenza del quadro generale della frase comporta l'inappropriatezza dei suggerimenti sia dal punto di vista sintattico che semantico.

2.2.2 Word Bigram

Il modello bigram espande il contesto e tiene conto della parola precedente a quella in cui è stato inserito il prefisso. Supponendo che l'utente abbia

digitato

$$\dots\ w_1\ w_{prefix}$$

il predittore che usa tale modello cerca tutte le parole w_2 del lessico che iniziano con w_{prefix} e poi verifica la probabilità $\mathbb{P}(w_2 \mid w_1)$. Infine suggerisce, tra quelle trovate, le k parole con probabilità (del bigram) più alte.

Un modello bigram quindi predice la parola w_i sfruttando l'informazione data dalla parola w_{i-1}. Questo modello è più accurato di un modello ad unigram.

Consideriamo un esempio per comprendere meglio il vantaggio di un modello bigram rispetto ad un modello unigram.

Supponiamo che l'utente abbia digitato:

"macchina costos"

e quindi si aspetta un suggerimento "costosa". Il modello unigram potrebbe suggerire "costoso", in quanto le probabilità di "macchina" e "costoso" sono indipendenti tra loro. Nel modello bigram invece la probabilità di avere "costoso" dopo "macchina", $\mathbb{P}(costoso \mid macchina)$ è inferiore alla probabilità di avere "costosa" dopo "macchina", $\mathbb{P}(costosa \mid macchina)$, e quindi la parola "costoso" non sarà suggerita, o meglio la parola "costosa" sarà suggerita prima della parola "costoso".

Come si può intuire un modello trigam sarà ancora più accurato di un

modello bigram, così come un modello bigram è migliore di un modello unigram.

La probabilità $\mathbb{P}(w_2 \mid w_1)$ di un bigram in un dato corpus è la probabilità di trovare la parola w_2 dopo la parola w_1. Questa probabilità si chiama *probabilità condizionale*[3] e viene calcolata come segue:

$$\mathbb{P}(w_2 \mid w_1) = \frac{\mathbb{P}(w_1,\, w_2)}{\mathbb{P}(w_1)} \tag{2.1}$$

dove $\mathbb{P}(w_1)$ è la probabilità che l'unigram w_1 occorra nel corpus ed è calcolata come

$$\mathbb{P}(w_1) = \frac{C(w_1)}{|C|}$$

e $\mathbb{P}(w_1, w_2)$ è la probabilità del bigram $w_1\, w_2$ ed è data dal numero delle sue occorrenze nel corpus normalizzata e viene calcolata come segue:

$$\mathbb{P}(w_1, w_2) = \frac{C(w_1, w_2)}{|C|}$$

Quindi sostituendo in (2.1), otteniamo:

$$\mathbb{P}(w_2 \mid w_1) = \frac{C(w_1,\, w_2)}{C(w_1)} \tag{2.2}$$

[3]Definizione. Sia B un evento tale che $\mathbb{P}(B) > 0$, si chiama probabilità di A condizionata a B (detta anche "probabilità di A dato B"): $\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A,B)}{\mathbb{P}(B)}$.

Ad esempio, supponiamo che l'utente abbia digitato:

"che ..."

la probabilità che la parola successiva sia "la" viene scritta come la probabilità condizionale $\mathbb{P}(la \mid che)$ e viene calcolata applicando (2.2) come segue:

$$\mathbb{P}(la \mid che) = \frac{C(che,\, la)}{C(che)}$$

ovvero il rapporto tra il numero delle occorrenze del bigram "che la" e il numero delle occorrenze dell'unigram "che".

Supponiamo che la parola "che" occorra 100 volte in un corpus di 1000 parole e che la parola "che" sia seguita dalla parola "la" 40 volte nel corpus, ovvero che il bigram "che la" occorre 40 volte nel corpus. Allora:

$$\mathbb{P}(la \mid che) = \frac{40}{100} = 0,4.$$

2.2.3 Word Trigram

Un modello trigram utilizza l'informazione data dalle parole w_{i-2} e w_{i-1} per predire la parola w_i. Il predittore che usa tale modello cerca tutte le parole del lessico che iniziano con il prefisso della parola w_i e poi verifica la probabilità $\mathbb{P}(w_3 \mid w_1,\ w_2)$ per ogni parola w_3. Infine suggerisce le k parole tra quelle trovate con probabilità più alta.

Supponiamo che l'utente abbia digitato:

"scrivo una ..."

mentre nel modello bigram o unigram suggerire "mela" potrebbe essere probabile, nel modello trigram la probabilità $\mathbb{P}(mela \mid scrivo\ una)$ è sicuramente meno alta della probabilità $\mathbb{P}(lettera \mid scrivo\ una)$ e quindi un predittore che usa il modello trigram suggerirà la parola "lettera" prima della parola "mela".

2.2.4 Modelli n-gram

È evidente che è possibile generalizzare i modelli precedenti a modelli n-gram, con $n > 3$. La predizione della parola successiva può essere vista come il tentativo di stimare la funzione di probabilità di avere una parola in una certa posizione della frase, data la sequenza di tutte le precedenti parole nel contesto (o history).

Questi dati tuttavia non sono in generale sufficienti per considerare ogni testuale history separatamente, ma è necessario un metodo che raggruppi le history simili per poter fornire predizioni sufficientemente accettabili per la parola successiva.

Una possibile soluzione per raggruppare le history è quella di adottare l'*Assunzione di Markov*, la quale prevede che soltanto le ultime $n-1$ parole della history influiscono sulla determinazione della parola successiva. Se riusciamo a costruire un modello in cui tutte le history con le stesse ultime

$n-1$ parole sono raggruppate nella stessa classe di equivalenza, otteniamo un modello di Markov del $(n-1)$-esimo ordine o un modello n-gram di parole (n-gram word model), come descritto nel paragrafo 2.3.2.

Quindi, stimiamo la funzione di probabilità $\mathbb{P}$ di ogni parola, w_n, considerando solo le $n-1$ parole precedenti: $w_1, w_2, \ldots, w_{n-1}$

$$\mathbb{P}(w_n \,|\, w_1,\, w_2, \ldots, w_{n-1})$$

ossia la probabilità che w_n sia la parola successiva nella frase formata dalle $n-1$ parole precedenti, $w_1, \ldots, w_{n-1}$.

Solitamente, vorremmo che n fosse sufficientemente grande da includere la maggior parte delle sequenze che potrebbero occorrere nella lingua in considerazione. Purtroppo, per grandi valori di n, il modello risulta essere più accurato e preciso ma ne consegue un numero troppo elevato di parametri da stimare.

Per questo motivo vengono usati gli unigram ($n = 1$), i bigram ($n = 2$) e i trigram ($n = 3$). Una possibile soluzione per ridurre il numero di parametri è quello di ridurre il valore di n, ma notiamo che gli n-gram non sono l'unico modo per formare classi di equivalenza della history; altri metodi si basano sul raggruppamento delle parole in classi semantiche, oppure sull'analisi della radice della parola, come lo stemming.

Stima della Probabilità di un n-gram

Cerchiamo di generalizzare il calcolo della probabilità per i word n-gram avendo osservato quello per i word bigram.

Consideriamo una frase composta da una sequenza di parole

$$w_1, w_2, \ldots, w_m$$

vogliamo calcolare l'occorrenza di ogni parola nella giusta posizione all'interno della sequenza come un evento indipendente, dobbiamo quindi calcolare la probabilità:

$$\mathbb{P}(w_1, w_2, \ldots, w_m) \tag{2.3}$$

La probabilità (2.3) può essere scomposta come una catena di probabilità, come segue:

$$\begin{aligned} \mathbb{P}(w_1, \ldots, w_m) &= \mathbb{P}(w_1)\mathbb{P}(w_2|w_1)\mathbb{P}(w_3|w_1, w_2) \ \ldots \ \mathbb{P}(w_m|w_1, \ldots, w_{m-1}) = \\ &= \prod_{k=1}^{n} \mathbb{P}(w_k|\, w_1, \ldots, w_{k-1}) \end{aligned}$$

Per stimare la probabilità

$$\mathbb{P}(w_k|\, w_1, \ldots, w_{k-1})$$

dobbiamo tenere conto dell'assunzione di Markov che asserisce che la proba-

bilità di una parola dipende solo dalla precedente, possiamo quindi approssimare:

$$\mathbb{P}(w_k|\, w_1, \ldots, w_{k-1}) \approx \mathbb{P}(w_k|\, w_{k-n+1}, \ldots, w_{k-1})$$

questa probabilità indica che la probabilità di una parola w_k in una frase date tutte le precedenti, può essere stimata come la probabilità della stessa parola date solo le precedenti n.

Riprendendo il risultato per il bigram e generalizzando al caso n-gram, otteniamo:

$$\mathbb{P}(w_k|\, w_{k-n+1}, \ldots, w_{k-1}) = \frac{C(w_{k-n+1}, \ldots, w_{k-1}, w_k)}{C(w_{k-n+1}, \ldots, w_{k-1})} \tag{2.4}$$

L'equazione (2.4) stima il rapporto tra la frequenza di una sequenza e la frequenza di un prefisso. Questo rapporto indica la frequenza relativa e viene spesso usato nella tecnica del Maximum Likelihood Estimation (MLE, stima di massima verosomiglianza) [31].

2.2.5 Smoothing

La stima Maximum Likelihood, vista nel precedente paragrafo, pone un problema per definire i parametri di training del modello n-gram. Questo problema viene chiamato "sparse data" ed è causato dal fatto che la stima di massima verosomiglianza, MLE, è basata su un particolare insieme di dati di addestramento. Per quanto tale insieme sia grande, rimane una porzione

finita e limitata. Perciò alcuni n-gram sono sicuramente assenti dal corpus e di conseguenza la loro probabilità sarà uguale a zero quando nella realtà non sarebbe così.

Questo inconveniente viene risolto con le cosidette tecniche di *smoothing* volte a garantire l'affidabilità delle frequenze degli n-gram che altrimenti avrebbero probabilità nulle. Di seguito vedremo un algoritmo di *smoothing*: l'interpolazione lineare.

Interpolazione Lineare

Come abbiamo visto nel precedente paragrafo nel modello n-gram e prendendo come esempio un modello trigram la probabilità che la parola che un utente intenda digitare sia w_3 date le parole precedenti w_1 w_2 è:

$$\mathbb{P}(w_3 \mid w_1\, w_2) = \frac{C(w_1, w_2, w_3)}{C(w_1, w_2)}$$

dove $C(w_1, w_2, w_3)$ è il numero di occorrenze del trigram w_1 w_2 w_3 nel corpus e $C(w_1, w_2)$ è il numero di occorrenze del bigram w_1 w_2 nel corpus.

Per ovviare al problema di "sparse data" (dispersione dei dati) si usa un modello che combina unigram, bigram e trigram trasformando il contributo nullo derivante da un modello unicamente a trigram in un valore di probabilità composto anche dal contributo relativo del modello bigram e unigram.

$$\mathbb{P}(w_3 \mid w_1\, w_2) = \alpha \cdot f(w_3 \mid w_1\, w_2) + \beta \cdot f(w_3 \mid w_2) + \gamma \cdot f(w_3)$$

dove

$f(w_3 \mid w_1\, w_2) = \frac{C(w_1, w_2, w_3)}{C(w_1, w_2)}$

$f(w_3 \mid w_2) = \frac{C(w_2, w_3)}{C(w_2)}$

$f(w_3) = \frac{C(w_3)}{|C|}$

$\alpha + \beta + \gamma = 1$

i termini α, β e γ sono dei pesi a somma unitaria ricavati dall'addestramento del sistema su un training set di testi. $|C|$ è il numero totale di occorrenze di tutte le parole nel corpus C.

2.3 Part-of-Speech Tagging

2.3.1 Introduzione

La predizione di parola determina quale parola potrebbe essere la successiva data la sequenza delle parole che la precedono. Basandosi però solo sulle statistiche delle parole non viene garantita la correttezza grammaticale della frase. Una possibile soluzione, per proporre all'utente suggerimenti grammaticalmente corretti, è quella di filtrare queste parole in base alla loro categoria sintattica. Il metodo più diffuso per assegnare ad ogni parola

la sua categoria sintattica si basa su classificazioni di tipo Part-of-Speech (POS).

Il processo di assegnamento di un'etichetta o *tag* a tutte le parole presenti in un testo, è chiamato *tagging* (dall'inglese etichettare, classificare), mentre un'applicazione che esegue tale compito è detta *tagger*. Tipicamente i tag indicano la categoria sintattica (sostantivo, verbo, articolo, ...). Il POS tagging consiste quindi nell'associare ad ogni parola del corpus la sua categoria sintattica e può essere ottenuto per codifica manuale o per lemmatizzazione semi automatica mediante un confronto delle parole con un dizionario.

Il significato della Part-of-Speech per il language processing è la grande quantità di informazioni che porta riguardo alla parola e quelle vicine ad essa. Questo è sicuramente vero per le principali classificazioni delle categorie (verbi, nomi), ma è anche vero per distinzioni più specifiche. Infatti, questi tag distinguono un aggettivo possessivo (mio, tuo, suo, loro) da un pronome personale (io, tu, egli). Sapere se una data parola è un aggettivo possessivo o un pronome personale ci può dire quali parole possono occorrere con maggiore probabilità nelle sue vicinanze. Ad esempio, possiamo osservare che è probabile che un aggettivo possessivo sia seguito da un sostantivo, mentre un pronome personale può essere seguito da un verbo.

Questo processo è utile nella modellizzazione del linguaggio per la predizione di parola poiché aggiunge valore al corpus, ma spesso viene usato anche in altre applicazioni linguistiche come il parsing, l'information ex-

tracting o la traduzione automatica. L'assegnamento automatico della POS gioca un ruolo molto importante negli algoritmi di disambiguazione delle parole (word-sense disambiguation) e nelle classi dei modelli n-gram. Infatti, i corpora che sono stati "marcati" o "taggati" dalla POS sono molto utili nella ricerca linguistica.

Esistono molti algoritmi di POS tagging tra cui:

- rule-based tagging (Sistemi a Regole);
- metodi probabilistici: HMM tagging (Hidden Markov Model tagging) e Maximum entropy tagging;
- transformation based tagging;
- memory based tagging;
- alberi decisionali.

Nel paragrafo 2.3.4 vedremo come funzionano i tagger basati su modelli markoviani, mentre le altre tecniche non verranno approfondite in quanto vanno oltre lo scopo di questa tesi. Per meglio capire il loro funzionamento introduciamo prima i modelli di Markov e gli Hidden Markov Model.

2.3.2 Modelli di Markov

I modelli di Markov sono modelli matematici usati per descrivere particolari processi stocastici. Un processo stocastico è una famiglia di variabili casuali

che dipendono dal tempo t:

$$\{X_t, t \in T\}$$

Le variabili casuali X_t sono definite sull'insieme X, detto spazio degli stati. Gli elementi $x_i \in X$ sono i valori che possono assumere le variabili casuali X_t e sono chiamati stati del sistema. Un processo stocastico è detto markoviano se le probabilità di transizione da uno stato all'altro dipendono unicamente dallo stato assunto dal sistema nell'istante precedente a quello considerato, ovvero, lo stato presente del sistema permette di conoscere lo stato futuro senza l'influenza della storia precedente (per questo motivo i processi markoviani sono detti "senza memoria").

Formalmente, sia:

$$X = (X_1, \ldots, X_t)$$

lo spazio degli stati e sia

$$S = \{S_1, \ldots, S_N\}$$

l'insieme degli stati del sistema, ovvero l'insieme dei valori che possono essere assunti dalle variabili in X.

Un processo stocastico è una ***catena di Markov*** se soddisfa entrambe le seguenti proprietà:

- la probabilità condizionata X_{t+1} dipende solo dalla probabilità X_t ed è indipendente dalle probabilità precedenti, matematicamente:

$$\mathbb{P}(X_{t+1} = S_k|X_1, \ldots, X_t) = \mathbb{P}(X_{t+1} = S_k|X_t) \tag{2.5}$$

- le probabilità sono stazionarie o costanti nel senso che dipendono dagli stati S_k e non dall'istante t in cui avviene la transizione né dal numero di transizioni, matematicamente:

$$\mathbb{P}(X_{t+1} = S_k|X_t) = \mathbb{P}(X_2 = S_k|X_1) \tag{2.6}$$

La proprietà (2.5) è detta *Orizzonte Limitato*, mentre la proprietà (2.6) è detta *Stazionarietà* o *Invarianza Temporale*.

2.3.3 Hidden Markov Model

Un ***Hidden Markov Model*** (Modello di Markov nascosto o HMM) è una catena di Markov i cui stati non sono osservabili direttamente, ovvero in cui l'esatta successione degli stati attraversati è nascosta o sconosciuta. Più precisamente:

- la catena ha un certo numero di stati;
- gli stati evolvono secondo una catena di Markov;

- ogni stato genera un evento con una certa distribuzione di probabilità che dipende solo dallo stato;
- l'evento è ossevabile ma non lo stato.

Formalmente un HMM è definito da una quadrupla (S, Π, A, B) di insiemi descritti come segue:

- $S = \{S_1, \ldots, S_N\}$ è un insieme finito di stati in cui S_i è lo stato i;
- π_i è al probabilità che S_i sia lo stato iniziale;
- a_{ij} è la probabilità della transizione $S_i \rightarrow S_j$;
- $b_i(k)$ è la probabilità di osservare l'emissione del simbolo k quando il modello si trova nello stato S_i.

  Inoltre:

- $\sum_i \pi_i = 1$, ovvero la somma delle probabilità degli stati iniziali deve essere uguale a 1;
- $\forall i, \sum_j a_{ij} = 1$, ovvero la somma delle probabilità di transizione uscenti da uno stato deve essere uguale a 1;
- $\forall i, \sum_k b_i(k) = 1$, ovvero la somma delle probabilità di emissione uscenti da uno stato deve essere uguale a 1.

Gli Hidden Markov Model sono usati nella modellizzazione statistica nei sistemi di riconoscimento vocale (ASR), ma anche in molte altre applicazioni

tra cui il riconoscimento dei testi manoscritti, in bioinformatica e principalmente nel trattamento automatico del linguaggio naturale ed in particolare nel Part-of-Speech tagging.

2.3.4 Markov Model Tagging

Il Markov Model Tagging è uno dei modelli più intuitivi ed è molto diffuso. Esso interpreta il problema di tagging in termini di teoria della probabilità. In particolare le sequenze di tag sono considerate come catene di Markov, descritte nella Sezione 2.3.2. In pratica, data una sequenza di parole $w_1, \ldots, w_n$, vogliamo trovare la sequenza dei tag $t_1, \ldots, t_{n-1}$ che ha la probabilità più alta, ovvero

$$t_1, ..., t_n = \arg \max_{t_1,...,t_n} \mathbb{P}(t_1, \ldots, t_n | w_1, \ldots, w_n) \tag{2.7}$$

Scomponiamo questa equazione in una serie di termini che sappiamo calcolare. Usando la legge di Bayes[4], possiamo riscrivere l'equazione (2.7) come:

$$t_1, ..., t_n = \arg \max_{t_1,...,t_n} \frac{\mathbb{P}(w_1, \ldots, w_n | t_1, \ldots, t_n) \cdot \mathbb{P}(t_1, \ldots, t_n)}{\mathbb{P}(w_1, \ldots, w_n)} \tag{2.8}$$

dove $\mathbb{P}(w_1, \ldots, w_n)$ sarà costante per tutte le combinazioni di tag (poiché la sequenza di parole non varia) e dunque non avrà alcun effetto sul calcolo di

[4] *Legge di Bayes*: siano A e B due eventi non trascurabili ($\mathbb{P}(A) > 0$ e $\mathbb{P}(B) > 0$) allora $\mathbb{P}(B|A) = \frac{P(A|B) \cdot \mathbb{P}(B)}{\mathbb{P}(A)}$

quale sequenza di tag è più probabile. Possiamo dunque semplificare (2.8) togliendo il denominatore, ottenendo:

$$t_1, ..., t_n = \arg\max_{t_1,...,t_n} \mathbb{P}(w_1, \ldots, w_n | t_1, \ldots, t_n) \cdot \mathbb{P}(t_1, \ldots, t_n) \tag{2.9}$$

Inoltre i Markov Model tagger stipulano due ipotesi:

1. La probabilità di una parola dipende solo dal proprio tag, non dai tag delle altre parole nella frase:

$$\mathbb{P}(w_1, \ldots, w_n | t_1, \ldots, t_n) \approx \mathbb{P}(w_1|t_1)\mathbb{P}(w_2|t_2) \ldots \mathbb{P}(w_n|t_n) \tag{2.10}$$

2. Assunzione di Markov del primo ordine, la quale prevede che la probabilità di un tag t_i dipende solo dal tag precedente t_{i-1}, come visto precedentemente è la proprietà di *Orizzonte Limitato* di una catena di Markov:

$$\mathbb{P}(t_1, \ldots, t_n) \approx \mathbb{P}(t_1|t_0)\mathbb{P}(t_2|t_1) \ldots \mathbb{P}(t_n|t_{n-1}) \tag{2.11}$$

Date queste assunzioni possiamo riscrivere (2.9) come segue:

$$t_1, ..., t_n = \arg\max_{t_1,...,t_n} \prod_{i=1}^{n} \mathbb{P}(w_i|t_i) \cdot \mathbb{P}(t_i|t_{i-1}) \tag{2.12}$$

dove $\mathbb{P}(t_i|t_{i-1})$ è dato dal prodotto delle probabilità di incontrare il tag t_i, se

il tag immediatamente precedente è t_{i-1}, e $\mathbb{P}(w_i|t_i)$ è dato dal prodotto delle probabilità, per ogni coppia parola/tag, di incontrare la parola w_i sapendo che il tag corrispondente è t_i.

In altre parole, $\mathbb{P}(t_i|t_{i-1})$ rappresenta la probabilità che il tag sia t_i se il tag precedente è il tag t_{i-1} (ad esempio, la probabilità che il tag della parola che sto analizzando sia verbo, se la parola che precede ha il tag pronome). Mentre $\mathbb{P}(w_i|t_i)$ rappresenta la probabilità che veda la parola w_i se tale parola ha il tag t_i (ad esempio, la probabilità che la parola che sto analizzando sia "studiato" se è un participio passato).

Avendo ridotto la formulazione probabilistica astratta del problema di tagging presentata da (2.7) a (2.12), ci troviamo con una serie di fattori che possiamo stimare sulla base del training corpus annotato. Le probabilità dei fattori del primo termine di (2.12) possono essere stimate usando la stima di massima verosomiglianza (o Maximum Likelihood Estimate, MLE) [26] e [27] come segue:

$$\mathbb{P}(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

Le probabilità dei fattori del secondo termine di (2.12) possono essere stimate, via MLE come:

$$\mathbb{P}(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

Estendiamo il Markov Model Tagging ai trigram. Nel tagger appena

descritto, assumevamo che la probabilità di un tag dipende solo dal tag precedente, ossia:

$$\mathbb{P}(t_1, \ldots, t_n) \approx \prod_{i=1}^{n} \mathbb{P}(t_i|t_{i-1})$$

Molti Markov Model tagger attuali usano una history più lunga assumendo che la probabilità del tag dipende dai due tag precedenti, ossia:

$$\mathbb{P}(t_1, \ldots, t_n) \approx \prod_{i=1}^{n} \mathbb{P}(t_i|t_{i-1}, t_{i-2})$$

e quindi la sequenza di tag con probabilità più alta è data da:

$$t_1, ..., t_n = \arg \max_{t_1,\ldots,t_n} \prod_{i=1}^{n} \mathbb{P}(w_i|t_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2})$$

dove, via MLE, possiamo stimare $\mathbb{P}(t_i|t_{i-1}, t_{i-2})$ come:

$$\mathbb{P}(t_i|t_{i-1}) = \frac{C(t_{i-2}, t_{i-1}, t_i)}{C(t_{i-2}, t_{i-1})}$$

Quando il training corpus non è sufficientemente ampio oppure il modello markoviano usato è di ordine maggiore (ad esempio trigam invece di bigram) molte sequenze di tag saranno estremamente rare o inesistenti nel training corpus e quindi sussiste un problema per stimare la probabilità (come già visto nella Sezione 2.2.5 per i word trigram che necessitano di *smoothing* a causa del "sparse data"). Anche in questo caso una possibile soluzione è

l'interpolazione lineare:

$$\mathbb{P}(t_i \mid t_{i-1}, t_{i-2}) = \alpha \cdot \mathbb{P}(t_i \mid t_{i-1}, t_{i-2}) + \beta \cdot \mathbb{P}(t_i \mid t_{i-1}) + \gamma \cdot \mathbb{P}(t_i)$$

richiedendo che $\alpha + \beta + \gamma = 1$, per assicurare che il risultato $\mathbb{P}$ sia sempre una distribuzione di probabilità.

2.4 Algoritmi di Predizione

Alcuni sistemi di predizione usano solo le sequenze di parole per effettuare predizione, altri invece impiegano n-gram di POS al posto delle parole, altri ancora sono sistemi ibridi che incorporano entrambi e altri ancora le combinano per ottenere un unico modello. In questa Sezione vedremo gli algoritmi usati da questi sistemi di predizione.

2.4.1 Predittore Sintattico (Solo Part-of-Speech Tag)

Nella word prediction usare la Part-of-Speech ha due principali vantaggi. Il primo vantaggio è dato dal fatto che essa prende in considerazione la sintassi della frase, poiché il tagger deve osservare le precedenti parole e le loro classificazioni (tag) per classificare la parola nella frase. Il secondo è dovuto al fatto che, poiché il modello trigram per le POS è molto più piccolo del modello trigram per le parole, il predittore sintattico può osservare le classificazioni delle due precedenti parole quando suggerisce le parole all'u-

tente, considerando un contesto più grande e quindi più significativo. Per predire la parola corrente, il predittore sintattico ha come input la seguente sequenza di parole e POS

$$\ldots \;\; w_{i-2}/t_{i-2} \;\; w_{i-1}/t_{i-1} \;\; w_{iprefix}$$

dove t_{i-1} e t_{i-2} sono le POS delle precedenti parole w_{i-1}, w_{i-2}. Per un word predictor predire la parola w_i significa riuscire a stimare la probabilità di avere questa parola nella posizione corrente dati i tag più probabili t_{i-1} e t_{i-2} delle due precedenti parole. Questa probabilità viene stimata come segue:

$$\mathbb{P}(w_i|t_{i-1}, t_{i-2}) = \sum_{t_i \in T(w_i)} \mathbb{P}(w_i|t_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2})$$

dove t_i è la POS tag di w_i e $|T(w_i)|$ è il set di tutti i possibili tag che possono essere assegnati a w_i. $\mathbb{P}(t_i|t_{i-1}, t_{i-2})$ è la probabilità del trigam di POS e $\mathbb{P}(w_i|t_i)$ è la probabilità condizionale di avere la parola w_i come parola corrente nella frase dato il tag t_i come sua classificazione. Quest'ultima probabilità può essere calcolata con la formula di Bayes come segue:

$$\mathbb{P}(w_i|t_i) = \frac{\mathbb{P}(t_i|w_i) \cdot \mathbb{P}(w_i)}{\mathbb{P}(t_i)}$$

dove $\mathbb{P}(w_i)$ è la probabilità della parola w_i e $\mathbb{P}(t_i)$ è la probabilità dei tag t_i. $\mathbb{P}(t_i|w_i)$ è la probabilità condizionale calcolata dal corpus per ogni parola w_i

e tutte le sue possibili classificazioni.

2.4.2 Tag and Words

Un altro modo per incorporare le statistiche dei tag POS nell'algoritmo di predizione è quello di stimare la probabilità di ogni parola w_i data la precedente parola w_{i-1}, il suo tag t_{i-1} e il tag POS della parola che la precede t_{i-2} [17, 18], ossia stimare la probabilità:

$$\mathbb{P}(w_i|w_{i-1}, t_{i-1}, t_{i-2})$$

Questa probabilità può essere stimata come segue:

$$\mathbb{P}(w_i|w_{i-1}, t_{i-1}, t_{i-2}) = \sum_{t_i \in T(w_i)} \mathbb{P}(w_i, t_i|w_{i-1}, t_{i-1}, t_{i-2}) \tag{2.13}$$

dove $\mathbb{P}(w_i, t_i|w_{i-1}, t_{i-1}, t_{i-2})$ viene calcolata come segue:

$$\mathbb{P}(w_i, t_i|w_{i-1}, t_{i-1}, t_{i-2}) = \mathbb{P}(w_i|w_{i-1}, t_i, t_{i-1}, t_{i-2}) \cdot \mathbb{P}(t_i|w_{i-1}, t_{i-1}, t_{i-2}) \tag{2.14}$$

In Figura 2.1 è mostrata una semplice rete Bayesiana, nella quale sono specificate le dipendenze condizionali dirette tra t_{i-2}, t_{i-1} , w_{i-1}, t_i e w_i.

Dobbiamo fare due assunzioni:

1. Usiamo l'ipotesi dell'indipendenza condizionale, come si può vedere nella rete Bayesiana, assumiamo che se il tag più probabile della parola

corrente è stato trovato dati i due tag precedenti, non è necessario sapere quali siano questi due tag. Quindi, $\mathbb{P}(w_i|w_{i-1}, t_i, t_{i-1}, t_{i-2})$ può essere scritta come $\mathbb{P}(w_i|w_{i-1}, t_i)$.

2. Assumiamo inoltre che il tag della parola corrente sia condizionalmente indipendente dalla parola precedente stessa, conoscendo il suo tag POS. Quindi, $\mathbb{P}(t_i|w_{i-1}, t_{i-1}, t_{i-2})$ può essere scritta come $\mathbb{P}(t_i|t_{i-1}, t_{i-2})$.

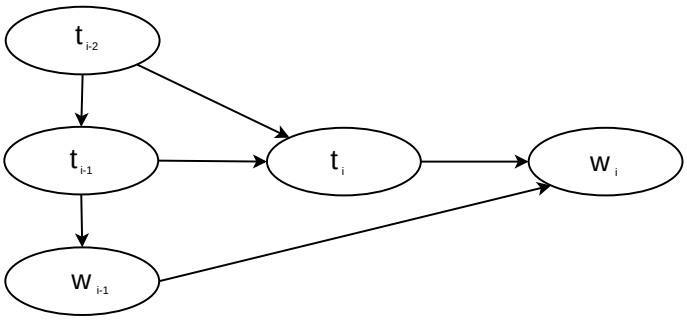

Figura 2.1: Rete Bayesiana che mostra le dipendenze tra parole e tag.

Come risultato di queste assunzioni e usando il teorema di Bayes, la probabilità in (2.14) può essere stimata come segue:

$$\begin{aligned}
\mathbb{P}(w_i, t_i|w_{i-1}, t_{i-1}, t_{i-2}) &= \mathbb{P}(w_i|w_{i-1}, t_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2}) \qquad (2.15)\\
&= \frac{\mathbb{P}(w_{i-1}, t_i|w_i) \cdot \mathbb{P}(w_i) \cdot P(ti|ti-1, ti-2)}{\mathbb{P}(w_{i-1}, t_i)}\\
&= \frac{\mathbb{P}(w_{i-1}|w_i, t_i) \cdot \mathbb{P}(t_i|w_i) \cdot \mathbb{P}(w_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2})}{\mathbb{P}(w_{i-1}, t_i)}
\end{aligned}$$

In accordo con la rete Bayesiana, si osserva che t_i e w_{i-1} non dipendono direttamente l'uno dall'altro, possiamo quindi sostituire il primo termine di

(2.15):

$\mathbb{P}(w_{i-1}|w_i, t_i)$ con $\mathbb{P}(w_{i-1}|w_i)$, ottenendo:

$$\frac{\mathbb{P}(w_{i-1}|w_i) \cdot \mathbb{P}(t_i|w_i) \cdot \mathbb{P}(w_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2})}{\mathbb{P}(w_{i-1}, t_i)} \tag{2.16}$$

A questo punto riscriviamo tramite il teorema di Bayes il primo termine di (2.16): $\mathbb{P}(w_{i-1}|w_i)$ come

$$\frac{\mathbb{P}(w_i|w_{i-1}) \cdot \mathbb{P}(w_{i-1})}{\mathbb{P}(w_i)}$$

ed anche il denominatore di (2.16): $\mathbb{P}(w_{i-1}, t_i)$ viene riscritto tramite Bayes come:

$$\mathbb{P}(t_i|w_{i-1}) \cdot \mathbb{P}(w_{i-1})$$

Sostituendo quindi in (2.16), si ottiene:

$$\frac{\mathbb{P}(w_i|w_{i-1}) \cdot \mathbb{P}(w_{i-1})}{\mathbb{P}(w_i)} \cdot \frac{\mathbb{P}(t_i|w_i) \cdot \mathbb{P}(w_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2})}{\mathbb{P}(t_i|w_{i-1}) \cdot \mathbb{P}(w_{i-1})}$$

e semplificando, si ottiene:

$$\frac{\mathbb{P}(w_i|w_{i-1}) \cdot \mathbb{P}(t_i|w_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2})}{\mathbb{P}(t_i|w_{i-1})}$$

Sempre in accordo con la rete Bayesiana, t_i e w_{i-1} non dipendono direttamente l'uno dall'altro possiamo quindi sostituire il denominatore $\mathbb{P}(t_i|w_{i-1})$

con $\mathbb{P}(t_i)$ e quindi la probabilità (2.14) può essere stimata dalla seguente formula:

$$\mathbb{P}(w_i, t_i | w_{i-1}, t_{i-1}, t_{i-2}) \approx \frac{\mathbb{P}(w_i|w_{i-1}) \cdot \mathbb{P}(t_i|w_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2})}{\mathbb{P}(t_i)}$$

tornando quindi alla (2.13), possiamo calcolare la probabilità di ogni parola con la seguente formula:

$$\begin{aligned}
\mathbb{P}(w_i, t_i | w_{i-1}, t_{i-1}, t_{i-2}) &= \sum_{t_i \in T(w_i)} \mathbb{P}(w_i, t_i | w_{i-1}, t_{i-1}, t_{i-2}) \\
&\approx \sum_{t_i \in T(w_i)} \frac{\mathbb{P}(w_i|w_{i-1}) \cdot \mathbb{P}(t_i|w_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2})}{\mathbb{P}(t_i)} \\
&= \mathbb{P}(w_i|w_{i-1}) \sum_{t_i \in T(w_i)} \frac{\mathbb{P}(w_i|w_{i-1}) \cdot \mathbb{P}(t_i|w_i) \cdot \mathbb{P}(t_i|t_{i-1}, t_{i-2})}{\mathbb{P}(t_i)}
\end{aligned}$$

2.4.3 Combinazione Lineare

L'algoritmo di combinazione lineare combina due modelli: i tag POS trigram e i word bigram. L'idea di base di questo metodo è che il predittore prima cerca di trovare il miglior tag di Part-of-Speech della parola corrente in accordo con le due precedenti POS. Successivamente cerca le parole che hanno la più alta probabilità di stare in quella posizione in accordo con il miglior tag trovato. A questo punto combina la probabilità del miglior tag trovato con la probabilità della parola, data la parola precedente. I due predittori, quello che predice il tag corrente in accordo con i due tag precedenti e quello

che usa i bigram di parole per trovare la parola più probabile, vengono combinati usando la combinazione lineare con un coefficiente $0 \leq \alpha \leq 1$,come segue:

$$\alpha \cdot \mathbb{P}(w_i|w_{i-1}) + (1-\alpha) \cdot \mathbb{P}(w_i|t_{cw}) \cdot \mathbb{P}(t_{cw}|w_{i-1}, w_{i-2})$$

dove $\mathbb{P}(w_i|w_{i-1})$ è la probabilità del bigram di parole (word bigram) e t_{cw} è il miglior tag per la parola w_i corrente nella posizione corrente e può essere trovato con la seguente formula:

$$t_{cw} = \arg\max_{t_{ij}} \{ \mathbb{P}(t_{ij}|w_{i-1}, w_{i-2}) \cdot \mathbb{P}(w_i|t_{ij}) \}$$

Una difficoltà importante sta nel determinare il valore di α: generalmente questo valore viene determinato sperimentalmente [18].

2.5 Adattamento e Apprendimento

Qualsiasi algoritmo il sistema di predizione usi, esistono diverse tecniche per rendere la predizione più accurata e più appropriata. I meccanismi di adattamento e apprendimento cercano di adeguare il modello del linguaggio allo stile e al lessico dell'utente per migliorare la qualità della predizione. Ci sono diversi approcci per incorporare questi meccanismi in un sistema di predizione di parola. Alcuni sistemi usano un *lessico adattabile* che aggiorna le frequenze delle parole in accordo con la scrittura dell'utente. Ad esempio, un meccanismo di predizione basato su un modello statistico del linguaggio

che memorizza il numero di occorrenze di unigram, bigram e trigram può essere dotato della capacità di modificare i conteggi delle occorrenze durante l'utilizzo del sistema di predizione. Il testo prodotto dall'utente verrà utilizzato per aggiornare i conteggi, imparando quali siano le parole preferite dall'utente.

Un altro esempio di apprendimento è costituito dall'occorrenza di nuove parole (parole che non occorrono nel training corpus). Queste nuove parole verranno inserite nel modello arricchendo il vocabolario e permettendo al predittore di suggerire parole precedentemente sconosciute. Un modo per aggiornare la statistica delle parole inserite dall'utente è quella di incrementare il numero di occorrenze ogni volta che una parola viene digitata. Un *lessico adattabile* può anche usare la *recency information*.

La *recency information* indica quanto recentemente una parola è stata usata in un dato contesto e quindi aumenta o diminuisce la probabilità che la parola venga usata un'altra volta in un contesto simile. Incorporare la *recency information* nel modulo di predizione può essere effettuato in diversi modi. Un modo per farlo è quello di usare un conteggio recency per ogni parola ed incrementare questo conteggio ogni qualvolta la parola venga usata nel contesto desiderato. Ogni documento o ogni sequenza di n parole nel documento può essere considerato come un nuovo contesto. Avremo bisogno di determinare quali parole dovranno essere selezionate in accordo con entrambe le frequenze, frequenze di base e frequenze recency.

Un altro modo per adattare le preferenze dell'utente è l'uso di un lessico orientato al soggetto, ossia una risorsa specifica per un determinato tipo di composizione testuale. Se l'utente deve comporre un documento tecnico caratterizzato da un insieme di vocaboli normalmente poco utilizzati e caratterizzati quindi da una bassa probabilità, il sistema suggerirà prima altre parole poco pertinenti al contesto specializzato. L'utente può comunicare al sistema di privilegiare i vocaboli specializzati e ottenere una predizione più mirata. Il vantaggio di avere diversi lessici è quello di restringere il range di parole scelte e quindi aumentare la probabilità di predire la parola appropriata.

Nel nostro sistema di predizione abbiamo implementato un meccanismo di adattamento e apprendimento basato su un dizionario personale dell'utente (descritto nel Capitolo 4). Questo dizionario, inizialmente vuoto, memorizza le parole dell'utente dinamicamente, durante la sessione di utilizzo, permettendo al predittore di suggerire quelle più usate. Inoltre, è stato usato e testato anche un lessico specializzato, un lessico radiologico (descritto nella Sezione 4.6.3).

2.6 Modelli Semantici

Il modello proposto in [30], è un modello integrato in cui le informazioni semantiche sono integrate con un modello n-gram al fine di migliorare la qualità e l'accuratezza della predizione. In questo modello vengono inizial-

mente selezionati un insieme di parole, specificamente dei sostantivi, semanticamente correlate (related words); ad esempio, "scuola" ha come parole correlate : "bambini, genitori, maestra, educazione". Queste parole formano la base di conoscenza semantica. Il processo di selezione di queste parole semanticamente correlate consiste in un primo stadio, nell'ordinamento (rank) del Pointwise Mutual Information (PMI)[5] delle parole co-occorenti e in seguito nell'identificazione della "correlazione semantica" (semantic relatedness) di queste parole tramite un filtro Lesk-like.

A questo punto la base di conoscenza viene usata per misurare l'associazione semantica tra le parole candidate per il suggerimento ed il contesto. In questo modo tali parole, che sono semanticamente approppriate al contesto, vengono messe in cima alla lista dei suggerimenti data la loro alta associazione con il contesto.

L'idea di base di questo modello è quella di misurare l'associazione semantica tra la parola da suggerire ed il contesto precedente, scegliendo le parole candidate con la più alta associazione per inserirle nella lista dei suggerimenti. Tale modello integrato unisce il modello semantico ed il modello n-gram[6], dove la predizione finale è data dalla combinazione dei due modelli. In particolare, le predizioni fornite dal modello n-gram sono filtrate e riorganizzate dal modello semantico.

[5]PMI la Mutua Informazione Puntiforme è una misura di associazione tra due parole usata in linguistica computazionale [8].

[6]Fazly e Hirst hanno costruito l'n-gram knowledge base (base di conoscenza semantica) e implementato il modello n-gram [18].

Vengono usate due basi di conoscenza: la base di conoscenza n-gram e la base di conoscenza semantica.

L'output della predizione finale è determinato dalla seguente formula:

$$\hat{w} = \arg\max \left(\log \mathbb{P}_{ngram}(w) + \log \left(1 + \lambda \cdot SA(w, CN) \right) \right) \tag{2.17}$$

dove:

$\hat{w}$ è una delle parole più probabili da predire (in realtà non viene preso un singolo $argmax$ ma le T parole con punteggio più alto da aggiungere nella lista dei suggerimenti lunga T).

Il contesto corrente CN è la sequenza di parole w_{i-3} , w_{i-2} , w_{i-1} che l'utente ha digitato.

$P_{ngram}(w)$ è la probabilità della parola w fornita dal modello n-gram.

$SA(w, CN)$ è l'associazione semantica tra la parola w e il contesto CN e

λ è il parametro usato per registrare il peso dell'associazione semantica, esso viene determinato sperimentalmente sul training corpus.

Se w non ha un'associazione semantica con il contesto CN allora SA vale 0 ed il modello di predizione integrato è determinato dal modello n-gram, altrimenti l'informazione fornita dal modello n-gram sarà usata insieme all'associazione semantica per determinare la lista dei suggerimenti per la parola data. Di seguito viene presentata una breve descrizione dell'algoritmo proposto. L'algoritmo riguarda un singolo ciclo di predizione. Se

l'utente non trova il suggerimento giusto nella lista dei suggerimenti e digita un nuovo carattere, un nuovo ciclo inizia con l'insieme dei suggerimenti ridotti di conseguenza.

I passi principali dell'algoritmo sono:

1. L'utente digita un prefisso nella posizione corrente, supponiamo sia "sc" per la parola scuola;

2. Il modello n-gram crea la lista dei suggerimenti per questo prefisso;

3. Per ogni parola trovata w del passo 2 calcola:

$$(\log \mathbb{P}_{ngram}(w) + \log (1 + \lambda \cdot SA(w, CN)))$$

4. Ordina il risultato in base al punteggio e mostra le T parole trovate all'utente;

5. L'utente decide se la parola si trova o meno nella lista dei suggerimenti.

Determinazione delle parole sementicamente correlate

Per ogni parola del dizionario viene determinato dal corpus l'insieme delle parole semanticamente correlate ad essa ed il grado di relazione di ogni correlato. Questa relazione viene usata come associazione semantica (SA) nell'equazione (2.17). L'informazione viene estratta dalla base di conoscenza semantica. Vediamo i passi di questo processo.

Per ogni parola w, di cui dobbiamo estrarre le parole semanticamente correlate, il *co-occuring words extractor* trova quelle parole co-occorrenti a w nella finestra definita come segue: per i sostantivi e i verbi che occorrono insieme, l'intera frase è considerata come contesto, perché la frase è l'unità dell'argomento e intuitivamente si prevede che i suoi sostantivi e verbi siano concettualmente correlati. D'altra parte, la finestra di testo per gli aggettivi è definita molto più ristrettamente: solo le cinque parole precedenti alla parola da predire. Questo perché soltanto le parole più vicine all'aggettivo sono correlate concettualmente alla parola da predire. Per esempio nella frase "*the prospectus gives a report on the students viewpoint and can be obtained from* ***individual*** *offices at some colleges of* ***higher*** *education*", gli aggettivi *individual* e *higher* si riferiscono solo ai loro sostantivi adiacenti piuttosto che agli altri sostantivi. E' corretto scrivere ***higher*** *education* ma non (nel caso di questa frase) ***higher*** *prospectus* o ***higher*** *offices*.

A questo punto le parole semanticamente correlate vengono selezionate grazie ai loro PMI (Pointwise Mutual Information) calcolato per le coppie di parole che occorrono insieme. Il PMI tra due parole che occorrono insieme viene calcolato come segue:

$$PMI(w_1, w_2) = \log_2 \frac{\mathbb{P}(w_1, w_2)}{\mathbb{P}(w_1) \cdot \mathbb{P}(w_2)} \tag{2.18}$$

dove $\mathbb{P}(w_1, w_2)$ è la frequenza delle parole (w_1, w_2) che occorrono insieme nel

corpus e $\mathbb{P}(w_i)$ è la frequenza della parola w_i nel corpus. Le parole con PMI più alto sono considerate automaticamente come parole legate alla parola da predire.

Le parole che occorrono insieme vengono ordinate in base al valore del loro PMI ed il numero esatto scelto sarà usato come parametro nell'algoritmo, queste parole vengono chiamate *seed words*. Per esempio le *seed words* della parola "scuola" includono: "matematica", "genitori", "maestra". Le parole rimanenti nella lista sono a questo stadio soltanto possibili parole correlate e vengono inviate al filtro di "correlazione" (relatedness filter)[7] per un'ulteriore identificazione di "correlazione" (relatedeness). Per esempio per la parola "scuola" queste parole includono: "bambini", "programma", "scienza".

Insieme ad ogni parola viene memorizzata la correlazione con la parola da predire, ossia, $Relatedness(w_i, w_j)$:

$$Relatedness(w_i, w_j) = \frac{C(w_i, w_j)}{C(w_i) \cdot C(w_j)}$$

dove $C(w_i, w_j)$ è il numero di occorrenze della coppia di parole (w_i, w_j) che occorrono insieme nel corpus e $C(w_i)$ è il numero di occorrenze della parola w_i nel corpus.

[7] Il metodo usato per il relatedness filter è chiamato Lesk-like, perché assomiglia all'algoritmo di Lesk per la word-sense disambiguation, ed è descritto in [30].

Associazione semantica con il contesto

Data questa base di conoscenza della "correlazione" semantica (semantic relatedness), possiamo ora calcolare l'associazione semantica della parola candidata, da predire, con il suo contesto semplicemente sommando il *Relatedness* di ogni coppia di parole formata dalla parola candidata con le parole del suo contesto.

Se $CN = \{ w_i \mid w_i$ è la parola (content word) nella frase$\}$ è il contesto e w è la parola candidata allora l'associazione di w con il contesto CN è calcolata come segue:

$$SA(w, CN) = Relatedness(w, w_i) \tag{2.19}$$

Se il contesto della parola, per esempio "costruzione", non è correlato alla parola candidata, supponiamo "scuola", allora il valore di *Relatedness* è 0. Di conseguenza se nessuna delle parole del contesto è correlata alla parola candidata, la parola candidata sarà considerata come non avente relazione semantica con il contesto. Vediamo un esempio.

Supponiamo che l'utente abbia inserito la seguente frase

"*Oats, salads and baked potatoes form the basis of three daily m* ".

Il modello n-gram proporrà un certo numero di parole candidate come per esempio: *market, media, marking, more, me, my, ... , meals,* ecc. A questo punto la parte semantica del modello integrato misurerà per og-

ni parola candidata l'associazione semantica con il contesto tramite l'equazione (2.19). Infine le due parti dell'informazione sono integrate tramite l'equazione (2.17).

Capitolo 3

Risorse Linguistiche

3.1 Introduzione

Avere a disposizione risorse linguistiche e strumenti che ne permettono la costruzione e la gestione è di fondamentale importanza per la progettazione e l'implementazione di applicazioni e soluzioni software per il linguaggio naturale.

La progettazione e l'implementazione del predittore di parola sviluppato per questo lavoro di tesi si basa sulla tecnologia Synthema per il trattamento del linguaggio naturale (*Lexical System Technology*^{TM}, o LST). In particolare è stato usato l'ambiente Synthema Lexical Studio che integra al suo interno il sistema di gestione di basi di dati lessicali, il costruttore dell'albero sintattico (Parser), l'editor dei dizionari e delle grammatiche ed il lemmatizzatore per la disambiguazione contestuale. Lexical Studio possiede

inoltre risorse che contengono le definizioni di morfologia, sintassi e semantica della lingua italiana, ed i dizionari generali e dei sinonimi. Inoltre è possibile modificare e testare le risorse come il linguaggio (morfologia, sintassi, semantica), il lessico (monolingua, bilingua, sinonimi e contrari) e la grammatica (struttura della frase, analisi errori, lemmatizzazione).

3.2 Il Sistema di Gestione di Basi di Dati Lessicali

Un modulo essenziale di Lexical Studio per applicazioni in linguaggio naturale è il Sistema di Gestione di Basi di Dati Lessicali (Lexical Data Base Management System, o LDBMS), originariamente sviluppato presso il Centro Ricerca IBM di Pisa, e successivamente perfezionato e ampliato da Synthema [34]. LDBMS è capace di gestire contemporaneamente più dizionari (monolingua, bilingua, sinonimi) e più lingue.

I dizionari possono essere anche molto grandi ed in generale i tempi di accesso non dipendono dalle dimensioni del singolo dizionario. Il sistema è stato appositamente disegnato e realizzato con architettura ABCD (**A** **B**asic **C**omputer **D**ictionary) al fine di ottimizzare la velocità di accesso alle informazioni lessicali, la flessibilità e la trasportabilità dei dizionari su diversi ambienti e piattaforme.

L'idea alla base del sistema è fornire un'indipendenza tra l'organizzazione delle parole e le regole dalle quali sono governate. Questa separazione è possibile poiché sono considerate due entità distinte: il **dizionario** e il **linguag-**

gio. Il primo è l'insieme delle parole che costituiscono un lessico, mentre il secondo è l'insieme di tutte le regole (morfologiche, sintattiche e semantiche) che governano una lingua. Questa distinzione è ragionevole in quanto solitamente lessici differenti di uno stesso linguaggio (ad esempio un lessico medico ed uno matematico) dipendono dalle stesse regole, che non avrebbe senso duplicare. Grazie a questa idea e alla sua realizzazione all'interno del sistema è possibile mantenere più dizionari regolati da uno stesso linguaggio.

Riassumendo le principali caretteristiche del LDBMS sono:

- l'indipendenza del sistema di gestione dall'applicazione;
- l'indipendenza del sistema di gestione dal linguaggio;
- l'indipendenza dal numero di dizionari e dalle dimensioni dei lessici;
- la capacità di memorizzare relazioni tra entità lessicali, ed informazioni sia sintattiche che semantiche.

3.3 Il Dizionario

Il dizionario è una risorsa molto importante per un predittore di parola e, in generale, per ogni sistema per il trattamento del linguaggio naturale. In particolare come vedremo nel Capitolo 4, è una delle risorse usate nel sistema di predizione sviluppato in questa tesi. Nel sistema attuale è disponibile un dizionario di base della lingua italiana: Italbase composto da circa 43.000 lemmi, 876.000 forme e 1.165.000 classificazioni.

3.3.1 Organizzazione Logica del Dizionario

Le informazioni nel dizionario hanno un formato tabellare e l'informazione contenuta ha la seguente struttura logica:

$lemma$[1] : (POS, $elenco\ feature$, $codice\ di\ flessione$, $codice\ di\ alterazione$)

La POS (Part-of-Speech) è la categoria sintattica, e può essere una delle nove categorie grammaticali previste per l'italiano: articolo, sostantivo, aggettivo, pronome, verbo, avverbio, preposizione, congiunzione, interiezione. Nel caso di lemmi omografi, vengono inseriti tanti lemmi per quante sono le diverse Part-of-Speech di quel lemma, ad esempio "porto" verrà inserito due volte con POS: verbo e sostantivo.

L' elenco feature (caratteristica) è l'insieme degli attributi che descrivono il lemma secondo specifici aspetti: forma grafica (iniziale maiuscola o tutto maiuscolo), caratteristiche sintattiche (verbo transitivo, intransitivo, ausiliare) ed altre informazioni funzionali, utili alla grammatica per la costruzione della frase, ad esempio FAM per i termini che individuano delle parentele familiari, MONTH per i mesi dell'anno.

Il codice di flessione (IRULE dall'inglese Inflection Rule) rappresenta le modalità con cui un lemma deve essere flesso. Nel caso in cui un lemma abbia più paradigmi di flessione, la definizione del lemma viene ripetuta

[1] viene definito *lemma* la parola di cui tratta ciascuna entrata di un dizionario.

tante volte quante sono i paradigmi di flessione.

Il codice di alterazione (ARULE dall'inglese Alteration Rule), in analogia al caso precedente, rappresenta le modalità con cui un lemma deve essere alterato, e si possono avere più alterazioni per uno stesso lemma. In italiano si possono alterare quasi tutti i sostantivi, i verbi e gli aggettivi.

Alcuni esempi di elementi nel dizionario sono:

abolizionista: NOUN FEAT(RARE CONC ANIM HUM) IRULE(9)

certamente: ADV FEAT(FREQ MODAL) ARULE(92)

raccolto: ADJ FEAT(EVAL PASTPART) IRULE(1) ARULE(69)

Consideriamo il lemma "abolizionista":

- è un sostantivo [POS=NOUN];
- è classificato come parola poco usata (RARE), concreta (CONC), il soggetto che descrive è animato (ANIM) e umano (HUM).[elenco feature];
- segue la flessione associata alla regola con codice 9 (quella di "artista", che flette in "artisti" e "artiste").[codice flessione];
- non ha alterazione.[codice alterazione]

3.3.2 La Struttura del Dizionario

Il dizionario è la componente del LDBMS più utilizzata, quindi la sua strutturazione e le sue caratteristiche hanno una forte incidenza su tutto il sis-

tema. L'architettura ABCD (A Basic Computer Dictionary) descrive sia la struttura logica del dizionario, sia le tecniche di memorizzazione virtuale. Essa permette di ottimizzare la velocità di accesso alle informazioni lessicali e la flessibilità e la trasportabilità dei dizionari su ambienti e piattaforme diverse. I meccanismi di ricerca delle informazioni all'interno del dizionario sono stati scelti e implementati in modo da ottenere tempi di risposta efficienti, infatti prevedono il minor numero possibile di passi logici e, allo stesso tempo, permettono il reperimento della massima quantità di dati.

Nel dizionario generale sono memorizzate tutte le forme[2] delle parole. Ad esempio, dato il lemma "coraggioso", si hanno forme declinate per genere e numero ("coraggioso", "coraggiosa", "coraggiosi", "coraggiose"); dunque nel dizionario sono presenti le quattro forme e non il lemma. Questo approccio è detto *non generativo*, a differenza dell'approccio *generativo* in cui il dizionario contiene solo i lemmi e una serie di regole di flessione ed alterazione permette di risalire al lemma partendo da una forma. L'approccio non generativo ha lo svantaggio di dover tenere in memoria una quantità di dati maggiore (nell'esempio di "coraggioso", quattro forme invece di un lemma), ma la ricerca nel dizionario è più veloce e più semplice da implementare (non c'è bisogno di applicare regole di flessione ed alterazione) e garantisce una maggiore correttezza dato che sono memorizzate solo le parole valide,

[2]Con il termine forma si intende un aspetto morfologico della parola, ovvero una sua variante lessicale, come la forma femminile, la forma plurale o la forma attiva e passiva di un verbo.

mentre un dizionario generativo deve possedere ed applicare regole di flessione ed alterazione (ad esempio, una regola che asserisce che "sorella" ha il plurale "sorelle", ma non ha il maschile "sorello").

Le forme sono organizzate in una Tree Word List (TWL), ovvero una doppia struttura ad albero che consente di identificare una parola come la composizione di una parte iniziale sinistra detta **radice** ed una parte finale destra detta **suffisso**. Tale organizzazione logica è rispecchiata da quella fisica, infatti l'albero delle radici (Lexicon TWL) e quello dei suffissi (Language TWL) sono memorizzati in due strutture dati distinte rispettivamente nel dizionario e nel linguaggio. Entrambi i componenti della TWL hanno un elevato grado di condivisione, tale per cui l'albero di tutte le forme, è estremamente compatto. Attualmente l'intero dizionario italiano è composto da circa 43.000 lemmi per 876.000 forme ed occupa meno di 14 Mb.

L'albero delle radici (Lexicon TWL), come possiamo vedere in Figura 3.1, è composto da nodi di transito (raffigurati a forma circolare) e da nodi terminali (raffigurati a forma ottogonale). I nodi terminali hanno un puntatore ad una struttura dati detta *Tabella dei Suffissi* che rappresenta la connessione tra la radice e tutti i possibili suffissi ad essa aggiungibili per ottenere tutte le forme. Ad ogni parola corrisponde una riga unica, puntata da un unico nodo terminale, che lega la parte sinistra a quella destra della parola. Ogni riga della tabella dei suffissi identifica una forma. Inoltre, tale tabella ha un puntatore ad un'altra struttura dati detta *Tabella degli*

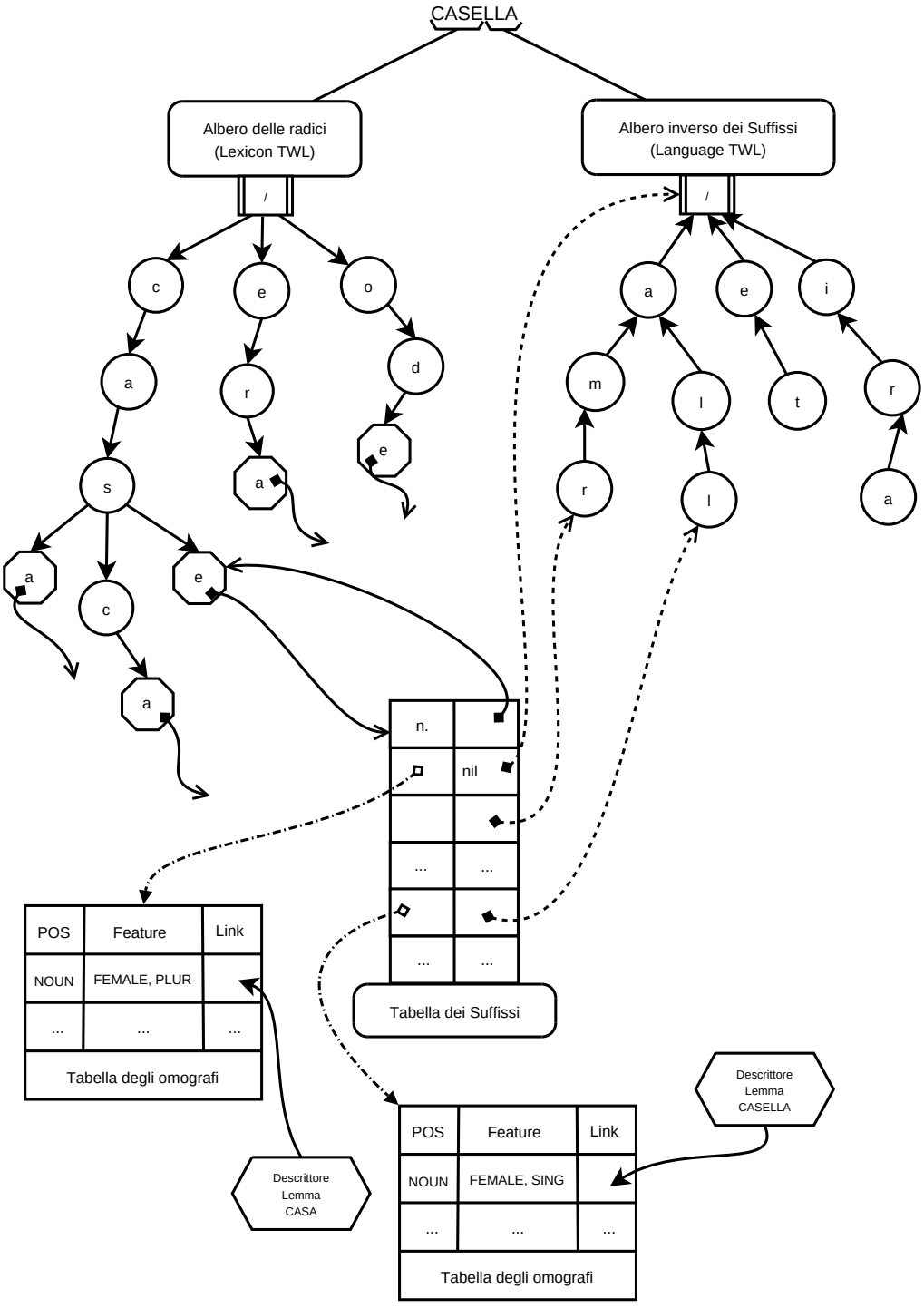

Figura 3.1: Tree Word List per il lemma "*casella*".

Omografi di ogni forma per determinare tutte le possibili classificazioni[3].

3.3.3 Generazione e Gestione dei Dizionari

Per poter essere utilizzati all'interno di Lexical Studio, i dizionari devono essere "generati". Tale operazione consiste nell'espandere i lemmi al fine di formare tutte le parole derivabili attraverso le regole previste nel linguaggio (regole di flessione, alterazione). La generazione del dizionario non avviene spesso ed è legata alle regole definite dal linguaggio di riferimento del dizionario. In Lexical Studio l'analizzatore sintattico, il parser, opera per riconoscimento e confronto delle parole analizzate con quelle generate. Le parole della lingua sono tutte presenti nei dizionari utilizzati dal parser al momento dell'analisi, garantendo prestazioni non ottenibili con i motori di riconoscimento morfologico operanti a tempo di esecuzione.

3.4 Il Linguaggio

Nel linguaggio sono definite le modalità di flessione, coniugazione, alterazione, le informazioni di tipo sintattico e semantico e le regole della grammatica di una lingua. Alcune di queste informazioni vengono usate ogni tanto in fase di generazione dei dizionari, mentre la grammatica sarà riferita ogni volta che avviene l'analisi sintattica di una frase.

[3]Classificare una forma significa determinare la categoria grammaticale, il lemma di provenienza nonché genere e numero.

3.4.1 Modalità di Flessione

Le regole di flessione (Inflection rule, IRULE) per aggettivi e sostantivi sono descritte in strutture apposite e organizzate in modo per cui ogni categoria riferisce un lemma-tipo. Per esso è indicata la completa flessione attraverso una regola che ha associato un codice di identificazione. Un esempio di regola di flessione per la declinazione dei sostantivi che seguono il lemma-tipo "artista" è riportato nella tabella seguente:

IRULE (9)
MALE FEMALE SING (*artista*)
MALE PLUR (*artisti*)
FEMALE PLUR (*artiste*)

Tabella 3.1: Regola di flessione per i sostantivi.

Le modalità di coniugazione dei verbi sono descritte tramite tabelle numerate indicanti la coniugazione di un verbo-tipo. Nella Tabella 3.2 sono mostrate le regole di coniugazione di un verbo regolare appartenente alla prima forma (*are*).

3.4.2 Modalità di Alterazione

Le modalità di alterazione (Alteration rule, ARULE) sono descritte in strutture apposite, in cui sono riportate tutte le possibili tipologie di alterazione previste da uno specifico lemma preso come modello di base. Attraverso l'espansione di questo lemma si ottengono le forme alterate, come il diminutivo,

CODE 001 ***amare***					
Indicativo Presente					
amo	ami	ama	amiamo	amate	amano
Indicativo Imperfetto					
amavo	amavi	amava	amavamo	amavate	amavano
Passato Remoto					
amai	amasti	amò	amammo	amaste	amarono
Futuro Semplice					
amerò	amerai	amerà	ameremo	amerete	ameranno
Congiuntivo Presente					
ami	ami	ami	amiamo	amiate	amino
Congiuntivo Imperfetto					
amassi	amassi	amasse	amassimo	amaste	amassero
Condizionale Presente					
amerei	ameresti	amerebbe	ameremmo	amereste	amerebbero
Imperativo					
—	ama	ami	amiamo	amate	amino
Participio Presente			Infinito		
amante	amanti		amare	amar	
Participio Passato				Gerundio	
amato	amata	amati	amate	amando	

Tabella 3.2: Regola di coniugazione per i verbi.

il vezzeggiativo o il superlativo. Nella Tabella 3.3 è mostrato un esempio di schema di alterazione, in particolare lo schema di alterazione per la formazione del diminutivo di tutte le parole che si declinano come l'aggettivo "piccolo".

ARULE(1) DIM BASE(*piccolo*)
MALE SING (*piccolino*)
FEMALE SING (*piccolina*)
MALE PLUR (*piccolini*)
FEMALE PLUR (*piccoline*)

Tabella 3.3: Regola di alterazione.

L'alterazione viene descritta in modo generico per tutte le flessioni, ma al momento della generazione vengono costruite solo le forme plausibili, ad esempio il diminutivo dell'aggettivo "bello" è espanso in "bellino", "bellina", "bellini" e "belline". Applicando la stessa regola di alterazione al sostantivo "macchina" vengono generate soltanto le forme corrette: "macchinina" e "macchinine", e non le due forme maschili.

3.4.3 Proprietà Sintattiche e Semantiche

Ad ognuna delle nove categorie grammaticali possono essere attribuite delle informazioni aggiuntive. Ogni lemma possiede proprietà che sono incluse esplicitamente nel dizionario (RARE, CONC, ANIM, ecc.) o individuate

al momento della generazione della forma flessa (FEMALE, PLUR, SUPR, PERSn, SUBJUNC, GERUND, ecc.).

Le proprietà sintattiche sono descritte in una tabella e attualmente descrivono fino a 128 caratteristiche diverse per ciascuna Part-of-Speech per un totale di circa 300 attributi. A livello semantico al momento non sono presenti informazioni. L'ampliamento di questo tipo di conoscenza può essere realizzato e gestito impiegando opportune informazioni da usare con le regole della grammatica, seguendo per esempio il modello descritto nella Sezione 2.6.

3.4.4 La Grammatica

I dizionari da soli non sono sufficienti ad individuare univocamente il ruolo grammaticale di una parola nella frase poiché indicano tutte le classificazioni sintattiche (POS) delle parole omografe, ma non contengono criteri di decisione per la scelta di una sola tra queste. L'eliminazione di tali ambiguità avviene ricorrendo ad un insieme di regole grammaticali definite secondo la sintassi del linguaggio LSR (**L**inguaggio per la **S**crittura di **R**egole) [39].

Tali regole grammaticali devono essere compilate e caricate in uno spazio di memoria opportuno. La fase di compilazione permette di controllare la correttezza formale delle regole. Solo a questo punto il parser può utilizzarle per l'analisi della frase. L'analisi della frase può far uso di due diversi tipi di grammatiche: la grammatica del dizionario e la grammatica generale del

linguaggio.

La Grammatica del Dizionario

La grammatica del dizionario è una grammatica di tipo specialistico ed è quindi associata al dizionario e non al linguaggio. Tale grammatica raggruppa le locuzioni ("a quattr'occhi", "stanco morto"), le espressioni composte ("Banca d'Italia", "bacino idrografico") e quelle idiomatiche proprie di una particolare lingua o di un dialetto ("fare la cresta", "prendere un granchio"). Il significato complessivo di queste espressioni non corrisponde alla semplice composizione del significato letterale delle singole parti: si parla di **M**ulti **W**ord **E**xpression (MWE), da cui il nome di *Grammatica MWE*.

CODE <0731>
PREP(BASE==‘‘*in*’’, TYPE==ARTFORM)
NOUN(BASE==‘‘*caso*’’, NUMB==SING)
CONJ(BASE==‘‘*che*’’)
$\Longrightarrow$ CONJ(=:(CONJ))

Tabella 3.4: Una Multi Word Expression (MWE).

La grammatica MWE ha lo stesso formalismo delle regole del linguaggio e secondo tale grammatica una sequenza di token (relativi a più componenti lessicali) che verifica particolari condizioni viene raggruppata in un singolo token. Un esempio di MWE è riportato nella Tabella 3.4.

L'espressione "in caso che", nell'esempio in Tabella 3.4, viene trasformata, in seguito a questa regola, in un unico token. Il perno della regola è la congiunzione "che"; la regola controlla che l'elemento perno sia preceduto dal sostantivo singolare "caso" e prima ancora dalla preposizione semplice "in". Se il pattern è verificato viene prodotto il nuovo token CONJ (congiunzione) a cui viene associata tutta la sequenza.

La Grammatica Generale

L'altra grammatica, distinta da quella delle MWE, è la grammatica generale del linguaggio, le cui regole possono essere suddivise in due gruppi: le *regole positive*, per il riconoscimento della struttura della frase, e le *regole negative*, per l'individuazione degli errori. Un esempio di regola positiva e di una negativa sono riportate rispettivamente nella Tabella 3.5 e nella Tabella 3.6.

CODE <P0015>
ADJ(GEND==GEND(NOUN), NUMB==NUMB(NOUN), ALTC==GOW||CASE(NOUN))NOUN() $\Longrightarrow$ CNOUN(=:(ADJ), PERS=PERS3, HEAD (NOUN), DET=(ADJ))

Tabella 3.5: Esempio di regola positiva.

Nella regola presentata in Tabella 3.5 NOUN () è il perno. Il token precedente al sostantivo perno è un aggettivo e deve concordare in genere e numero con il perno, ed inoltre ci deve essere concordanza in alterazione

(ALTC). La funzione CASE(NOUN) può essere VOWEL, CONS, o SPUR a seconda che l'iniziale del sostantivo sia rispettivamente una vocale (acqua), consonante pura (ragazzo) o consonante impura (scuola). Quindi il pattern specifica che l'aggettivo precedente il sostantivo dovrà avere l'attributo ALTC uguale a GOWVOWEL (go with vowel, ad esempio bell'), GOWCONS (bel) o GOWSPUR (bello) in concordanza con l'iniziale del sostantivo: in caso di "s" o "g" vengono considerate le prime due lettere: "il sale","lo stabile", "il giorno", "lo gnomo".

Se il pattern è verificato allora viene prodotto un nuovo token CNOUN (Composite Noun), di livello più alto, che specifica il ruolo di terza persona del sintagma.

Un esempio di regola negativa è riportato in Tabella 3.6.

CODE <p0090>
CONJ(ID=MAIN, BASE==‘‘*ma*’’)
CONJ(BASE==‘‘*però*’’)
$\Longrightarrow$ ECONJ(ERR==24)

Tabella 3.6: Esempio di regola negativa.

Questa regola produce un errore nel caso in cui si presenti la erronea costruzione "ma però". Il perno è la congiunzione "però" e, se è preceduto dalla congiunzione "ma", si verifca il pattern segnalando l'errore.

3.5 Il Parser

Il software per la gestione delle risorse può essere visto come composto da due moduli principali: il gestore dei dizionari ed il parser sintattico che esegue diverse fasi successive di analisi della frase ed è basato sul linguaggio.

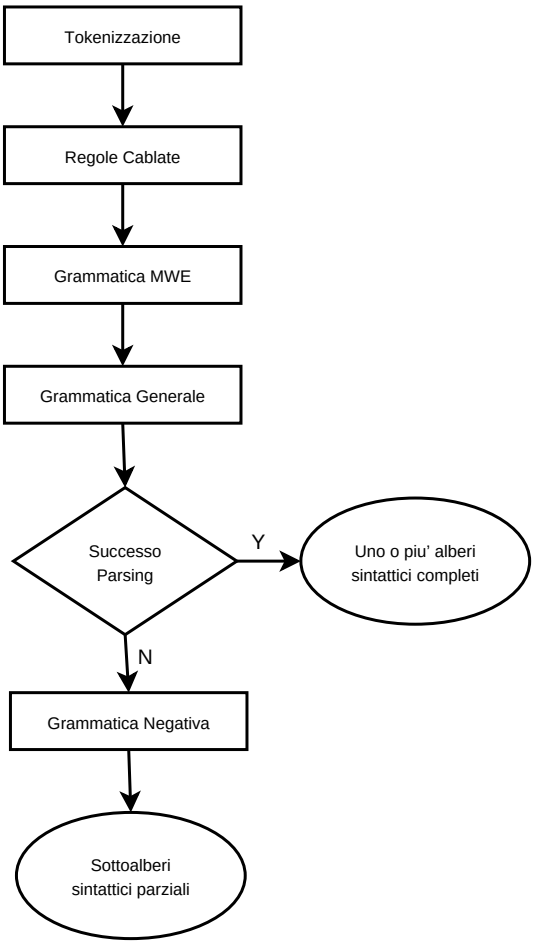

Figura 3.2: Passi Principali del Parser Synthema

In Figura 3.2 sono riassunte le fasi che esegue il Parser sintattico.

1. *Tokenizzazione*: viene eseguita la scansione e la suddivisione in token

(ossia unità lessicali minime) della frase in input e in seguito vengono classificati attraverso l'uso del dizionario. Questa operazione porta alla costruzione di una catena di token formata da tutte le classificazioni, anche multiple, delle parole che compongono la frase.

2. *Regole Cablate*: corrispondono a regole "deterministiche" che completano la costruzione o l'esclusione di un singolo token. Ad esempio il sostantivo femminile singolare "ancora" può avere anche il ruolo di avverbio temporale: una regola appropriata verifica il ruolo della parola rispetto alle parole adiacenti ed eventualmente trasforma il sostantivo in avverbio (infatti, nel caso in cui la parola precedente sia un articolo quella successiva non può essere un avverbio). In questo modo il token viene specializzato riducendo il numero delle sue possibili classificazioni.

3. *Grammatica MWE*: in questa fase la catena di token viene analizzata dalla grammatica del dizionario, individuando le espressioni polilessicali (MWE) e realizzando così una seconda riduzione del numero di elementi.

4. *Grammatica Generale*: successivamente viene applicata la grammatica generale del linguaggio per effettuare la disambiguazione delle classificazioni. Viene, in questo modo, prodotto un albero sintattico che cerca di riportare tutte le foglie ad un'unica radice. La chiusura dell'albero

rappresenta la corretta individuazione dei legami tra i token e di conseguenza il riconoscimento della struttura grammaticale ammessa per la frase. Se la frase è ambigua si possono ottenere più alberi sintattici.

5. *Grammatica Negativa*: Se lo scopo dell'analisi (parsing) è l'individuazione o la correzione degli errori, l'analizzatore (parser) prevede la possibilità di attivare la grammatica negativa che produrrà nuovi token relativi a quelli ritenuti discordanti e segnalerà le costruzioni risultate errate.

3.6 La Lemmatizzazione

La lemmatizzazione è l'operazione di ricondurre ogni parola di un testo alla forma base o entrata di dizionario. Consiste nel riunire tutte le forme sotto il rispettivo lemma, intendendo per *lemma* ciascuna parola-titolo o parola-chiave di un dizionario e per *forma* ogni possibile diversa realizzazione grafica di un lemma. Sono esempi di forme *mangerò, mangeremmo, mangiaste, mangi*: il lemma cui ciascuna di queste forme è riducibile è *mangiare*, ovvero la parola che appare come entrata sui dizionari.

Nell'ambiente Lexical Studio la lemmattizzazione viene effettuata analizzando l'albero sintattico risultato del parser. In particolare l'operazione viene realizzata associando ad ogni forma flessa una coppia definita come segue:

[(***sestupla***), (***lemma di origine***)]

La "*sestupla*" identifica una rappresentazione di sei informazioni, in particolare la prima informazione identifica la Part-of-Speech della parola (che è la stessa del lemma). Ad esempio un risultato della lemmatizzazione per la parola "bambina" è riportato in Figura 3.3.

$$\underbrace{\texttt{bambina [(S C F S}}_{forma}\ \underbrace{\texttt{M S), (bambino)]}}_{lemma}$$

Figura 3.3: Classificazione del sostantivo "*bambina*".

La parola "bambina" è una forma flessa del lemma "bambino". La sestupla (SCFSMS) denota che la parola "bambina" è un sostantivo (POS = S), comune (C), di genere femminile (F) e singolare (S). Le due ultime informazioni riguardano il lemma di origine, in questo caso "bambino", che infatti è di genere maschile (M) ed è singolare (S).

La classificazione del sostantivo "donna" è invece:

donna [(S C F S F S), (donna)]

In questo caso il lemma di origine è "donna" e quindi femminile (F) singolare (S). Altri esempi sono mostrati nella Tabella 3.7, dove sono riportati alcune classificazioni di verbi.

- "è" viene classificato come un verbo (POS = V), ausiliare essere (E),

è	[(V E N 3 I N), (essere)]
aveva	[(V A N 3 I I), (avere)]
entrato	[(V F M S P P), (entrare)]

Tabella 3.7: Classificazione di verbi.

neutro (N), terza persona singolare (3), indicativo (I), presente (N), avente lemma "essere";

- "aveva" è un verbo (POS = V), ausiliare avere (A), neutro (N), terza persona singolare (3), indicativo (I), imperfetto (I), avente lemma "avere";

- "entrato" è un verbo (POS = V), intransitivo con ausiliare essere (F), maschile (M), singolare (S), participio (P), passato (P), avente lemma "entrare".

Lo schema generale delle informazioni contenute nelle sestuple per le diverse POS è riassunto nella Tabella 3.8.

Nella Tabella 3.8 "N" sta per *neutro*, infatti nella lingua italiana gli aggettivi e i pronomi possessivi non contengono informazioni di genere relative al referente, a differenza della lingua inglese, ad esempio ("his", "her").

Nel caso in cui il pronome possessivo sia soggetto e singolare, il campo "3/6" indica che la persona del verbo deve essere terza, altrimenti, se il pronome è plurale, deve essere la sesta.

	POS	1	2	3	4	5	6
1	sostantivi	S	PROPRIETÀ	GENERE	NUMERO	*genere*	*numero*
2.1	aggettvi	G	PROPRIETÀ	GENERE	NUMERO	0	0
2.2	agg. possessivi	G	PROPRIETÀ	GENERE	NUMERO	N	*persona*
3.1	pronomi	N	PROPRIETÀ	GENERE	NUMERO &	0	0
3.2	pr. personali				PERSONA	*f. sintatt.*	0
3.3	pr. possessivi				3 / 6	N	*persona*
4	verbi	V	PROPRIETÀ	GENERE	NUM & PERS	MODO	TEMPO
5	articoli	R	PROPRIETÀ	GENERE	NUMERO	0	0
6	preposizioni	P	PROPRIETÀ	GENERE	NUMERO	0	0
7	avverbi	A	PROPRIETÀ	0	0	0	0
8	congiunzioni	C	PROPRIETÀ	0	0	0	0
9	punteggiatura	@	PROPRIETÀ	0	0	0	0
10	interiezioni	I	0	0	0	0	0
11	unknown	U	0	0	0	0	0

Tabella 3.8: Schema di corrispondenza “POS / sestupla”.

Il significato dei valori contenuti nella Tabella 3.8 è il seguente:

- PROPRIETÀ: indica un’informazione aggiuntiva della forma che specifica ulteriormente la categoria;
- GENERE: può essere M, F o N a seconda che la forma sia, rispettivamente, maschile, femminile o neutra (ovvero valida sia per il maschile che per il femminile, ad esempio “docente” o “insegnante”);
- NUMERO: viene specificato S, P o I a seconda che la forma sia singolare,

plurale o indefinita (ovvero valida sia per il singolare che per il plurale, ad esempio "blu");

- NUMERO & PERSONA: oltre agli elementi previsti nel campo NUMERO, alcuni pronomi e verbi possono assumere un valore compreso tra 1 e 6 che corrisponde ad una combinazione di persona e numero (1 indica la prima persona singolare e 6 la terza plurale);

- *genere* (*del lemma*): nel caso di sostantivi e aggettivi, questo campo può assumere gli stessi valori del campo GENERE, mentre per i pronomi è prevista una diversa tipologia a seconda del caso;

- *numero* (*del lemma*): è l'ultimo elemento della sestupla per i sostantivi e può assumere gli stessi valori del campo NUMERO per il lemma;

- *persona* (*del lemma*): può assumere gli stessi valori del campo NUMERO & PERSONA(ovvero un numero tra 1 e 6 a seconda della persona del lemma);

- *funzioni sintattiche*: contiene valori per classificare la funzione della forma all'interno della frase: soggetto, oggetto, complemento indiretto, ecc.;

- MODO E TEMPO: nel caso di verbi, gli ultimi due elementi della sestupla permettono di identificare il modo (indicativo, congiuntivo, con-

dizionale, imperativo, participio, gerundio o infinito) ed il tempo (presente, futuro, passato o imperfetto).

Il significato degli ultimi due elementi della sestupla sono legati al primo elemento della medesima. Infatti questi due dati:

- per sostantivi, aggettivi possessivi e pronomi possessivi esprimono informazioni sul lemma di origine;
- per i pronomi personali il quinto elemento indica la funzione sintattica, mentre il sesto è nullo;
- per i verbi il quinto e il sesto carattere indicano rispettivamente il modo ed il tempo della coniugazione verbale;
- per le altre classificazioni gli ultimi due elementi sono nulli.

La Tabella 3.9 mostra alcuni esempi di parole con la rispettiva categoria grammaticale e l'informazione aggiuntiva. I dati contenuti in tale tabella sono stati estratti dall'analisi di alcuni testi lemmatizzati.

3.6.1 La Grammatica Statistica

La Grammatica statitistica è un'altra importante componente disponibile in Lexical studio ed è ispirata al modello teorico della **catene di markov**. Tale risorsa si basa sul concetto di "tripla" corrispondente alla definizione di "trigram", ossia una sequenza di tre sestuple che identificano la Part-of-Speech. Le triple sono particolarmente significative nel progetto realizzato

	Categoria Grammaticale	Informazione Aggiuntiva	Esempi
1	S = sostantivo	C = default	orchestra, stabilimento, clienti.
		H = persona	Enrico, Anna, Francesco
		T = città	Parigi, Verona, Algeri
		ecc.	
2	V = verbo	T = transitivo	misurare, svolgere, sposare.
		I = intransitivo	esistere, nascere, andare.
		ecc.	
3	R = articolo	D = determinativo	il, la, lo, l', i.
		I = indeterminativo	un, un', una, uno.
4	G = aggettivo	G = possessivo	suo, sua, loro, mio.
		N = numerale	milione, otto, 1980.
		I = indefinito	tanti, poche, altro.
		ecc.	
5	A = avverbio	T = temporale	ora, subito, ancora.
		L = di luogo	quil, ovunque, là
		C = comparativo	più, meno, come, meglio, così.
		ecc.	
6	C = congiunzione	C = coordinativa	e, ma, anche, o.
		S = subordinativa	che, quando, infatti
7	P = preposizione	S = semplice	di, a, da, in, con, su
		C = default	dei, della, nell', alla, sul.
8	N = pronome	P = personale	ci, lui, si.
		R = relativo	cui, quale, che, chi.
9	I = interiezione	0 = default	bravo, grazie, bene.
10	@ = punteggiatura	0 = default	"!", ";, "?"
11	U = unknown	0 = default	parole sconosciute

Tabella 3.9: Tabella di lemmatizzazione.

per questa tesi. Durante il progetto abbiamo riorganizzato la grammatica statistica in modo più efficiente. Di seguito descriviamo lo stato precedente di tale risorsa.

Le triple erano state estratte da un corpus giornalistico sufficientemente ampio e vario, infatti i testi sono stati selezionati tali da trattare diversi argomenti, scritti da diversi autori e contenuti in quotidiani, settimanali e mensili. Tale corpus è stato lemmatizzato con il parser. Il risultato della lemmatizzazione è stato poi analizzato per una revisione manuale. Tale revisione risulta onerosa ma indispensabile al fine di ottenere corpora corretti e disambigui. I corpora correttamente lemmatizzati sono infatti indispensabili per il successivo processo di generazione delle triple.

I corpora corretti sono stati poi analizzati per individuare gli "oggetti lessicali" che occorrono spesso insieme, al fine di individuare il modo in cui vengono più frequentemente costruite le frasi. In particolare sono state estratte le sestuple e create le triple di sestuple rispettando l'unità sintattica minima (la frase). Ad esempio, a partire da una frase composta da quattro parole, in cui la quarta è il simbolo di fine frase, possiamo ottenere due triple di sestuple. Indicando con w_i la i-esima parola e con $POS(w_i)$ la corrispondente sestupla, otteniamo le seguenti triple:

$$[POS(w_1), POS(w_2), POS(w_3)]$$

$$[POS(w_2), POS(w_3), POS(w_4)]$$

Ad ogni tripla estratta è stato, in seguito, attribuito un valore statistico

in base alla frequenza con cui i tre oggetti lessicali comparivano insieme. Le triple erano poi state memorizzate in una struttura dati ad albero (una TWL come quella per il dizionario).

Un'esempio di tripla molto probabile è:

[(pronome personale),	(verbo transitivo),	(articolo)]
io	*mangio*	*la*...(mela)
lei	*veste*	*il*...(bambino)

Tabella 3.10: Esempi di tripla.

La Grammatica statistica a triple permette, quindi, di effettuare la lemmatizzazione di un testo sfruttando le informazioni relative alle occorrenze di sequenze di parole all'interno di un dato corpus linguistico e non mediante l'analisi grammaticale e neanche ad un riferimento alla sintassi.

Capitolo 4

Il Progetto

In questo Capitolo presenteremo il lavoro effettuato per questa tesi, ovvero l'estensione di un predittore di parola. Inizieremo con una descrizione del percorso che è stato seguito nello sviluppo del progetto, poi entreremo in dettaglio nella descrizione delle innovazioni introdotte.

4.1 Descrizione del Progetto

L'obiettivo principale di questo lavoro di tesi è quello di sviluppare un sistema di supporto per velocizzare l'attività di scrittura al computer e minimizzare il numero di digitazioni.

Un tale obiettivo può essere interessante per qualsiasi tipo di utente, ma nel caso di utenti disabili diventa fondamentale per aumentare l'indipendenza, la capacità di comunicazione e in definitiva per migliorare la vita e l'inserimento sociale.

Il lavoro realizzato in questa tesi si è sviluppato in una prima fase di analisi del modulo di predizione esistente, in particolare abbiamo effettuato dei test per capire quali erano i difetti di tale sistema e quale era la capacità predittiva di tale modulo in termini di percentuale di caratteri risparmiati. Successivamente ci siamo dedicati alla definizione di strategie di correzione degli aspetti che non risultavano soddisfacienti.

In una terza fase abbiamo riorganizzato alcune risorse linguistiche e ne abbiamo create delle nuove.

Successivamente ci siamo concentrati sulla fase implementativa dell'algoritmo di predizione integrando le nuove risorse nel modulo di predizione e sperimentando diversi modelli statistici del linguaggio con l'obiettivo di usare nuovi approcci alla ricerca dei completamenti delle parole così da rendere più accurati i suggerimenti.

Infine nell'ultima fase abbiamo effettuato diversi test al fine di validare il lavoro effettuato verificandone la qualità della predizione.

Inoltre, abbiamo sperimentato il predittore su un corpus linguistico diverso dall'italiano di base, ovvero su un dizionario specialistico di radiologia. Per fare ciò è stato necessario creare nuove risorse linguistiche estratte da una grande raccolta di referti radiologici. Abbiamo integrato queste nuove risorse nel modulo di predizione. Infine abbiamo effettuato dei test per verificare la qualità della predizione con questo nuovo corpus radiologico.

4.2 Il Predittore: Architettura del Sistema

Il predittore di parola ha tre principali componenti, come mostrato in Figura 4.1: l'interfaccia utente, il modulo di predizione e le risorse linguistiche. Il motore di predizione è il cuore del modulo di predizione il quale controlla la comunicazione con l' interfaccia utente, tiene traccia dello stato della predizione e delle parole digitate. Ad ogni digitazione, predice i suggerimenti sotto forma di una lista di parole assicurando la concordanza sintattica con il contesto della frase.

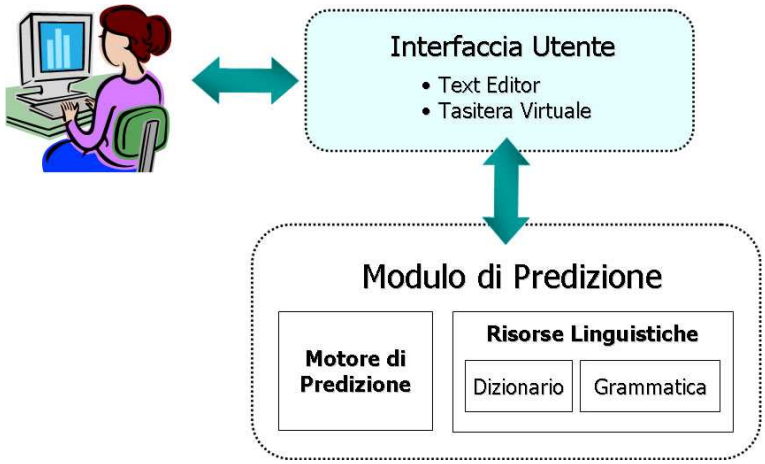

Figura 4.1: Architettura del Sistema.

Le funzionalità del modulo di predizione come la concordanza sintattica e la copertura del lessico sono forniti dal modello del linguaggio statistico basato su un modello n-gram di Part-of-Speech e di parole fornite dalle risorse linguistiche. Durante la fase di sviluppo è stata usata l'interfaccia di

test *Indovino*, un programma di videoscrittura che permette di sperimentare la scrittura assistita dal computer (come mostrato in Figura 4.2).

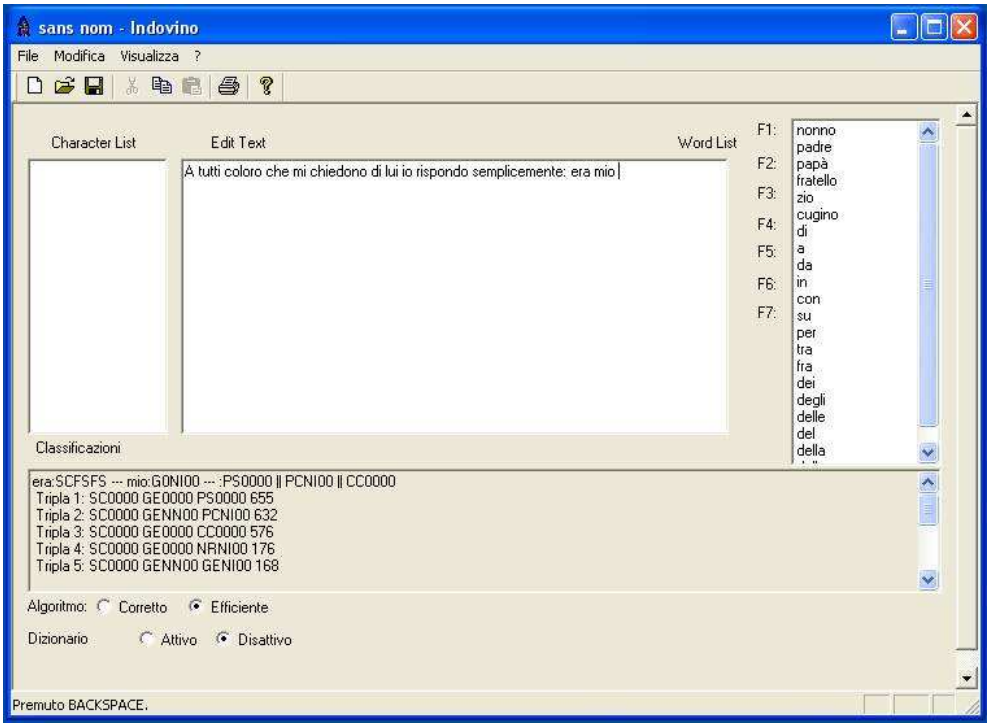

Figura 4.2: Interfaccia di test Indovino in funzione.

Il modulo di predizione detto *LxsPrd*, utilizza i dati provenienti dall'utente (caratteri digitati), dalle risorse linguistiche per generare la lista di completamenti da proporre all'utente attraverso Indovino, che interagisce proprio con questo modulo 4.3.

Come mostrato in Figura 4.2, al centro si trova la finestra di composizione (Edit Text) e a destra la lista di completamenti suggeriti (Word List). In basso, si trova la finestra di controllo delle classificazioni delle parole digitate ed i corrispondenti POS trigram più probabili. Nell'esempio, la frase finiva

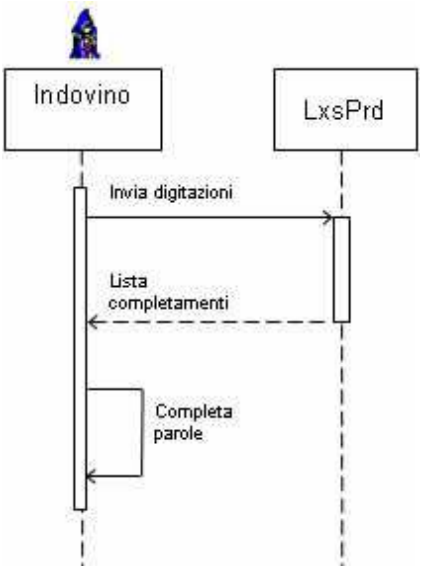

Figura 4.3: Interazione fra l'interfaccia utente e il modulo di predizione.

con "era mio nonno", il suggerimento giusto è in cima alla lista.

4.3 Realizzazione del Progetto

Prima di iniziare il lavoro per questa tesi abbiamo analizzato il modulo di predizione precedente [14], in particolare abbiamo effettuato diversi test sia per misurarne le capacità predittive in termini di caratteri risparmiati sia per valutarne le funzionalità. Alcune caratteristiche non erano soddisfacienti. In particolare abbiamo perfezionato la fase di cancellazione dei caratteri; migliorato la gestione dei POS trigram, ovvero la scelta di quelli più probabili; e modificato la gestione dei suggerimenti. Inoltre, la percentuale di caratteri risparmiati è risultata essere nei test effettuati, mediamente del 14%.

Alla luce dei risultati ottenuti, abbiamo deciso di modificare il sistema per renderlo più efficace in termini di predizione della parola, ovvero aumentare il numero di digitazioni risparmiate. Una volta che abbiamo iniziato a definire e stutturare il progetto ci siamo resi conto della necessità di adeguare e ampliare le risorse linguistiche. Inizialmente abbiamo riorganizzato le risorse linguistiche e successivamente ne abbiamo create di nuove.

In particolare abbiamo inziato col modificare la grammatica statistica a triple (POS trigram) che assicura la concordanza sintattica delle forme lessicali, descritta nel Capitolo 3. Infatti i POS trigram contenevano molte triple poco utili per la predizione: per esempio le triple contenenti i segni di punteggiatura risultano inutili per predire una virgola o un punto, in quanto in termini di riduzione dei caratteri digitati non si ha alcun risparmio. Inoltre il grande numero di POS trigram aumenta il tempo di ricerca del suggerimento.

Per questi motivi le triple sono state rigenerate dal corpus di testi usati per l'addestramento (testi lemmatizzati). Questa nuova lista di triple ignora i segni di punteggiatura considerandoli come separatori di parola, allo stesso modo dello spazio.

Inoltre, a questa nuova risorsa di POS trigram è stato applicato un *algoritmo di unificazione* per ridurne il volume e rendere la ricerca dei suggerimenti più generica rispetto a genere e numero. Un primo tipo di unificazione

è la *generalizzazione*. Vediamo un esempio per capirne il funzionamento. Supponiamo che l'utente abbia digitato:

"la penna di"

Dopo "la penna di" è possibile trovare un articolo indeterminativo maschile ("uno", volendo scrivere ad esempio "uno studente") o femminile ("una", volendo scrivere ad esempio "una scrittrice"). È pertanto opportuno suggerire sia il maschile che il femminile.

Le triple "nome comune-preposizione-articolo indeterminativo maschile" e "nome comune-preposizione-articolo indeterminativo femminile" sono state unite in una sola: "nome comune-preposizione-articolo indeterminativo" e le loro probabilità sono state sommate: la nuova tripla risulta così effettivamente più probabile.

Un secondo tipo di unificazione è la *parametrizzazione*, ossia quella effettuata su triple che concordano sempre in genere e numero. Prendendo ad esempio la frase "il mio gatto", grammaticalmente le tre parole concordano sempre. Infatti dopo "la mia" si può scrivere solo "gatta" e non "gatto", lo stesso vale per il plurale "i miei", seguito sempre da "gatti" e mai da un singolare. In questo caso la tripla è detta parametrica: le quattro triple possibili "articolo determinativo maschile singolare-aggettivo possessivo maschile singolare-nome comune maschile singolare", "articolo determinativo maschile plurale-aggettivo possessivo maschile plurale-nome comune maschile plurale", "articolo determinativo femminile singolare-aggettivo pos-

sessivo femminile singolare-nome comune femminile singolare", "articolo determinativo femminile plurale-aggettivo possessivo femminile plurale-nome comune femminile plurale" sono state accorpate in una sola: "articolo determinativo parametrico-aggettivo possessivo parametrico-nome comune parametrico".

Alla nuova tripla è stata assegnata una probabilità data dalla somma delle quattro probabilità delle triple di origine, così che se l'utente digita "la mia", il predittore determina che il parametro dell'articolo e dell'aggettivo è il femminile singolare e dunque propone solo nomi comuni femminili singolari. Unificando tutti i casi strutturati nei modi suddetti il numero delle triple è stato ridotto da 76.000 a 19.000, con una riduzione del 75%.

L'unificazione ha portato due benefici: il più evidente è dato dall'efficienza, infatti la riduzione del numero delle triple tra cui cercare ha aumentato la velocità della ricerca dei suggerimenti. Il secondo e più importante beneficio è dato dall'aumento del keystroke saving[1] dovuto alla maggiore genericità della ricerca dei suggerimenti e al riordinamento delle triple.

4.3.1 Nuove Risorse Linguistiche

Per migliorare la predizione di parola sono state create nuove risorse linguistiche: i *POS bigram*, i *word bigram* con POS, che chiamiamo: *word bigram taggati* ed i *Word Unigram*.

[1] Il keystroke saving è la percentuale delle digitazioni risparmiate, descritto nel Capitolo 5.

Un POS bigram è una coppia di tag di Part-of-Speech che appaiono in sequenza. Ad esempio: “io ti amo” viene lemmatizzata come:

io[NPN1S1(io)] ti[NPN2S2(tu)] amo[VTN1IN(amare)]

ossia, pronome personale prima persona, pronome personale seconda persona e verbo presente indicativo prima persona. I tre tag di POS vengono divisi in due POS bigram : “NPN1S1 NPN2S2” e “NPN2S2 VTN1IN”.

I POS bigram sono stati generati dal corpus di testi lemmatizzati usati per l’addestramento.

L’aggiunta ai meccanismi di predizione dei Word Bigram taggati, è la principale innovazione apportata al predittore. I Word Bigram taggati sono le coppie di parole presenti nel corpus con la Part-of-Speech relativa alla seconda parola. Ad esempio la frase “il mio cane” è composta dai due word bigram “il mio” e “mio cane”. I due word bigram taggati sono:

“il mio GGMSN1” e “mio cane SCMSMS”

dove GGMSN1 sta per aggettivo possessivo Maschile Singolare e SCMSMS sta per Sostantivo Comune Maschile Singolare.

Sono state create due liste separate di word bigram taggati: una formata da tutte le coppie di parole e una formata solo dalle coppie di parole di inizio frase, così da avere un modello apposito per predire le seconde parole di inizio frase. Tuttavia, per la parola iniziale della frase, è presente una

terza lista, composta da Word Unigram, ovvero da singole parole ordinate per frequenza.

I dati descritti sopra sono stati estratti dai testi lemmatizzati di addestramento grazie a diversi programmi scritti in linguaggio *Perl* sviluppati appositamente.

I word bigram taggati sono stati organizzati in una tabella hash, una struttura dati associativa che ad una "chiave", la prima parola, fa corrispondere un "valore", la seconda parola e relativa POS. La tabella hash è stata usata perchè permette un accesso più veloce rispetto ad un file di testo; inoltre è versatile e semplice da modificare e quindi ottima per sperimentare diversi approcci di predizione (cambiare l'insieme dei word bigram taggati, cambiare le probabilità dei word bigram taggati) e vedere qual è il migliore.

Inoltre abbiamo introdotto le liste PRICQ per le preposizioni, gli articoli, le interiezioni, le congiunzioni ed i numerali. Vedremo più avanti come e quando esse vengono usate.

Il dizionario generale utilizzato è lo stesso descritto nel paragrafo 3.3, integrato nella libreria linguistica; esso classifica le parole digitate dall'utente e fornisce suggerimenti aggiuntivi in ordine alfabetico: viene usato quando gli altri meccanismi di predizione non riescono a suggerire parole valide.

Inoltre, è stato creato un dizionario personale composto dalle parole digitate più spesso dall'utente, descritto nella sezione 4.5.

4.4 Statistiche

Come descritto nel Capitolo 2, i modelli statistici ad n-gram esprimono la probabilità a priori di sequenze di parole assumendo che una sequenza di parole W appartiene al linguaggio con probabilità $\mathbb{P}(W)$. Descrivono quindi la probabilità che la parola w_n sia preceduta dalla sequenza w_{n-2}, w_{n-1}.

4.4.1 Modello n-gram per i Tag di Part-of-Speech

La predizione della categoria sintattica, la Part-of-Speech, successiva si ottiene con un modello statistico n-gram basato sulle Catene di Markov. La probabilità della Part-of-Speech è data dalla seguente formula:

$$f(t_i, t_{i-1}, t_{i-2}) = \begin{cases} \mathbb{P}(t_i|t_{i-1}, t_{i-2}) & \text{se } \mathbb{P}(t_i|t_{i-1}, t_{i-2}) > \theta \\ \mathbb{P}(t_i|t_{i-1}) & \text{altrimenti} \end{cases}$$

dove $\mathbb{P}(t_i, t_{i-1}, t_{i-2})$ è la probabilità della tripla (POS trigram) e $\mathbb{P}(t_i, t_{i-1})$ è la probabilità del POS bigram.

Questa probabilità significa che quando il contesto della frase è lungo meno di tre parole o quando non ci sono triple che permettono di fare la predizione (non esistono triple per tutte le possibili combinazioni di POS), la predizione della categoria sintattica successiva si basa sui POS bigram.

4.4.2 Modello n-gram per le Parole

La probabilità dei word bigram taggati viene calcolata usando un modello word bigram (come quello che abbiamo visto nella Sezione 2.2.2) **esteso**. Il modello word bigram introdotto nella Sezione 2.2.2 stima la probabilità di una parola data la precedente: $\mathbb{P}(w_i|w_{i-1})$

Abbiamo esteso questo modello aggiungendo l'informazione riguardante la POS. Ovvero, ad ogni word bigram viene associata la Part-of-Speech della seconda parola (ad esempio: "di un RIMS00 279", dove "RIMS00" (articolo maschile singolare), è la Part-of-Speech assegnata a "un" e "279" è la frequenza di tale bigram nel corpus). La Part-of-Speech serve a classificare la parola senza ricorrere al dizionario e quindi permette di risparmiare tempo evitando alcune disambiguità che invece il dizionario potrebbe produrre; inoltre il dizionario non fornisce la frequenza di ogni parola che invece è necessaria per stimare le probabilità e calcolare la combinazione lineare che descriveremo nel paragrafo 4.6.2.

Il word bigram (w_{i-1}, w_i) è stato esteso a (w_{i-1}, w_i, t_i), dove t_i è la POS di w_i.

4.5 Dizionario Personale con Autoapprendimento

Il Dizionario personale è una ulteriore risorsa da cui il predittore può ricavare i suggerimenti per l'utente. Nel Dizionario personale possono essere inserite

fino a 3000 parole, questo limite massimo è stato scelto in base al numero medio di parole del lessico di una persona.

Le parole che saranno inserite dovranno appartenere alle seguenti categorie sintattiche: verbi, nomi, aggettivi, avverbi. Sono infatti queste le categorie di parole che maggiormente caratterizzano il lessico di una persona, per quanto in italiano siano spesso più frequenti articoli e congiunzioni.

La principale caratteristica del dizionario personale è quella di apprendere automaticamente le parole più utilizzate: inizialmente il dizionario personale è vuoto e man mano che l'utente digita, il predittore classifica le parole e, se appartengono alle quattro categorie sopra dette le memorizza in una tabella hash (come quella dei word bigram taggati).

Poiché la dimensione del dizionario è limitata a 3000 parole, si è deciso di usare un algoritmo di sostituzione che usa il meccanismo LRU (Least Recently Used) nel quale le parole che sono state utilizzate meno di recente vengono cancellate per registrare a loro posto le ultime parole digitate. In base a quanto recentemente una parola è stata digitata le si assegna un peso per inserirla nella lista di predizione tramite la combinazione lineare descritta nel paragrafo 4.6.2.

4.6 L'Algoritmo di Predizione

L'algoritmo di predizione si basa sulla combinazione lineare, che descriveremo più avanti in questo Capitolo, di due modelli linguistici, uno basato

sulle classificazioni delle parole e l'altro sulle parole stesse. Esso determina quali saranno le parole più probabili che l'utente intende digitare e presenta questo insieme di parole sottoforma di una lista di suggerimenti.

In questa Sezione presenteremo il funzionamento dell'algoritmo e il modello del linguaggio su cui si basa.

4.6.1 Funzionamento Generale dell'Algoritmo

L'algoritmo di predizione ha un diverso comportamento a seconda dello stato in cui si trova, ovvero: predizione della prima parola, predizione della seconda parola o predizione della terza parola e successive. Per meglio comprendere la logica dell'algoritmo riportiamo il modello seguito in Figura 4.4.

Per predire la prima parola che l'utente intende digitare, quindi dopo aver inserito il primo carattere, si usa un modello word unigram. Il predittore propone all'utente tutte le parole di inizio frase più probabili che iniziano con il carattere digitato. Una volta inserita la prima parola, questa viene classificata, ossia gli viene assegnata una POS.

Per predire la seconda parola, il predittore cerca i POS bigram più probabili che iniziano con la POS assegnata. A questo punto le seconde parole proposte nella lista dei suggerimenti, che potrebbero completare la digitazione in corso, vengono prese dai word bigram taggati di inizio frase. Le parole trovate vengono ordinate tramite la combinazione lineare delle probabilità di POS bigram più probabili e le probabilità delle parole.

Per predire la terza parola e le succesive, la ricerca delle parole viene effettuata nei word bigram taggati (non di inzio frase) e la ricerca della POS più probabile viene effettuata tra le triple. Se il POS trigram non esiste, la ricerca della POS viene effettuata tra i POS bigram. Se i suggerimenti trovati non sono sufficienti, vengono cercate altre parole provenienti dal dizionario e dalle liste PRICQ. Queste liste contengono: preposizioni, articoli, interiezioni, congiunzioni e numerali e vengono usate anche quando la POS più probabile predetta appartiene ad una di queste categorie lessicali. Le parole trovate vengono ordinate combinando linearmente le probabilità dei POS trigram con le probabilità dei word bigram.

L'aggiunta dei word bigram taggati al predittore FastType ha provocato un rovesciamento dell'approccio al completamento basato sul dizionario generale: i suggerimenti, infatti, non vengono più presi dal dizionario mettendo nella lista di predizione tutte le parole, in ordine alfabetico, che potrebbero completare la digitazione in corso; vengono invece ricavati dai word bigram taggati. Data una parola, il predittore inserisce nella lista di predizione tutte le parole che solitamente la seguono (ad esempio dopo "correre", il sistema propone "a", "via", "più", "velocemente" e così via), ordinandole in base alla loro probabilità. Quindi, le parole vengono proposte in ordine decrescente di probabilità.

Ovviamente, man mano che l'utente digita i caratteri, i suggerimenti vengono filtrati. Ad esempio, se dopo "correre" l'utente digita "v", nella

lista restano solo “via”, “velocemente” e le altre parole che iniziano per “v”.

Se nessuno dei suggerimenti provenienti dai word bigram taggati è accettato dall’utente, il predittore provvede a recuperarne altri dal dizionario generale, in ordine alfabetico.

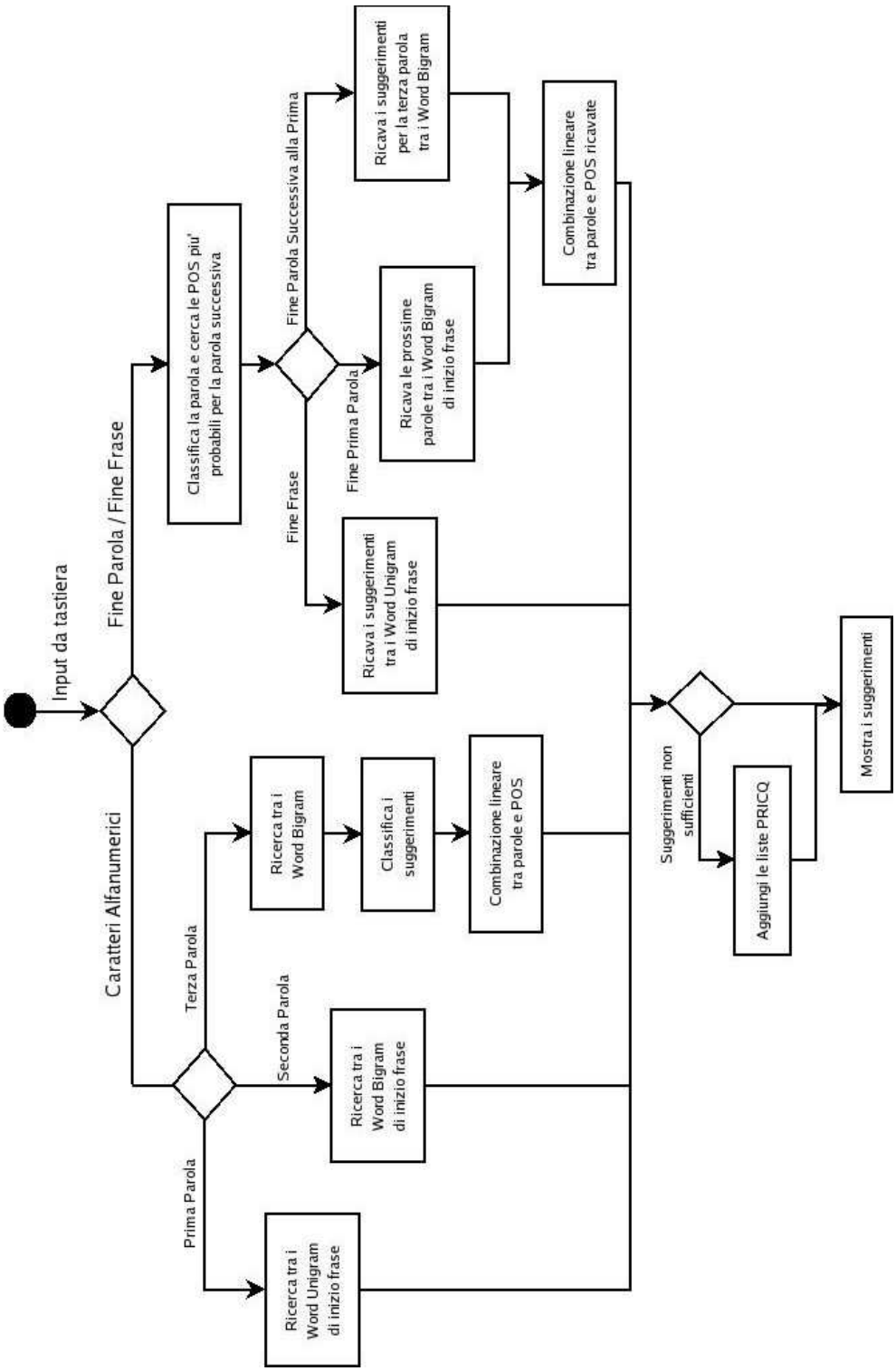

Figura 4.4: Diagramma di interazione generale del predittore FastType.

4.6.2 Combinazione Lineare

Come già detto il modello del linguaggio utilizzato è un modello n-gram di parole e POS, esteso con un modello unigram per le parole di inizio frase. Questi modelli vengono usati per stabilire il ranking di tutte le parole che l'utente potrebbe digitare. Il Ranking è il meccanismo che ordina i suggerimenti nella lista di predizione in base alla loro probabilità e non all'ordine alfabetico. All'utente sono mostrate le prime L parole, dove L è lunghezza della lista di suggerimenti. Il rank, posizione nella lista, di una parola è dato dalla sua probabilità. Questa probabilità è ricavata dalla combinazione lineare di due componenti, provenienti rispettivamente dalla Grammatica Statistica Generale (POS trigram) e dai word bigram.

L'algoritmo di combinazione lineare usato è un'estensione di quello descritto nella Sezione 2.4.3, e combina due modelli: POS Trigram (o POS Bigram) e word bigram taggati.

Il primo, POS Trigram, cerca e trova le tre POS più probabili nella posizione corrente date le due POS precedenti, se il contesto della frase è lungo meno di tre parole o se non ci sono triple che permettono di fare la predizione (non esistono triple per tutte le possibili combinazioni di POS), la predizione della categoria sintattica successiva si basa sui POS Bigram.

Il secondo modello, word bigram taggati, trova le parole più probabili per la posizione corrente data la parola precedente (la ricerca viene effettuata nella lista di word bigram taggati). I due modelli vengono combinati in

modo che la parola corrente abbia una probabilità S calcolata come segue:

$$S = \alpha \cdot \mathbb{P}(w_i|w_{i-1}, t_i) + \beta \cdot f(t_i, t_{i-1}, t_{i-2}) \tag{4.1}$$

dove $\mathbb{P}(w_i|w_{i-1}, t_i)$ è la probabilità del word bigram e

$$f(t_i, t_{i-1}, t_{i-2}) = \begin{cases} \mathbb{P}(t_i|t_{i-1}, t_{i-2}) & \text{se } \mathbb{P}(t_i|t_{i-1}, t_{i-2}) > \theta \\ \mathbb{P}(t_i|t_{i-1}) & \text{altrimenti} \end{cases} \tag{4.2}$$

è la probabilità del POS trigram. La formula (4.2) indica che, nel caso in cui la probabilità del POS trigram sia minore di una certa soglia θ, si usa la probabilità del POS bigram. α e β sono i coefficienti della combinazione lineare e la loro somma deve essere uguale a 1 ($\alpha + \beta = 1$). I valori di α e β sono stati determinati sperimentalmente (vedi Capitolo 5).

Oltre alla combinazione lineare dei due fattori POS trigram e word bigram taggati, abbiamo esteso il modello in modo che tenesse conto anche delle parole del dizionario privato (descritto nella Sezione 4.5). La probabilità S della parola corrente definita in (4.1) diventa:

$$S = \alpha \cdot \mathbb{P}(w_i|w_{i-1}, t_i) + \beta \cdot f(t_i, t_{i-1}, t_{i-2}) + \gamma \cdot \mathbb{P}(w_i) \tag{4.3}$$

dove α, β e γ sono tre coefficienti a somma unitaria ($\alpha + \beta + \gamma = 1$) e $\mathbb{P}(w_i)$ è la probabilità della parola ricavata dal dizionario personale.

4.6.3 Applicazione del Modello ad un Corpus Specialistico

Al fine di valutare il funzionamento del predittore con altri lessici lo abbiamo testato con un corpus specialistico medico-radiologico. Poiché la copertura del lessico di tale corpus rispetto alle risorse linguistiche presenti (dizionario generale, word unigram, word bigram) risultava essere molto bassa, ovvero molte parole risultavano sconosciute, abbiamo deciso di creare nuove risorse linguistiche medico-radiologiche. Tali risorse sono state estratte da un insieme di testi ottenuti da referti radiologici forniti dall'Ospedale di Pisa.

Le risorse linguistiche create sono quelle necessarie al predittore, ossia i word unigram (radiologici) di inizio frase e i word bigram taggati (radiologici). Per poter disporre dei word bigram taggati è stata necessaria una prima fase di lemmatizzazione, mentre le i POS trigram usate sono quelle descritte precedentemente.

L'algoritmo di predizione segue la stessa descrizione logica descritta precedentemente, ma se le parole provenienti dalle nuove risorse non sono sufficienti viene usato un dizionario radiologico per completare la lista dei suggerimenti.

4.6.4 Esempio di Funzionamento del Predittore

Durante la fase di implementazione e di testing è stata usata l'interfaccia grafica Indovino. Questa interfaccia permette di capire il funzionamento del

predittore mostrando incrementalmente la scelta dei POS trigram.

Riportiamo qualche immagine per mostrare il funzionamento del predittore sviluppato. Supponiamo, che l'utente voglia scrivere la frase "mia madre ha preparato una torta".

Come mostrato in Figura 4.5, dopo aver inserito il primo carattere "M", all'utente viene proposta una lista di parole che iniziano con questa lettera. Essendo nel caso del primo carattere della prima parola, il predittore propone una lista di parole ordinate per probabilità decrescente. Notiamo in Figura 4.5 che nella lista dei suggerimenti (Word List) la parola "mia" si trova in quarta posizione. Assumiamo che l'utente abbia selezionato il suggerimento: la situazione, a questo punto, è quella mostrata in Figura 4.6.

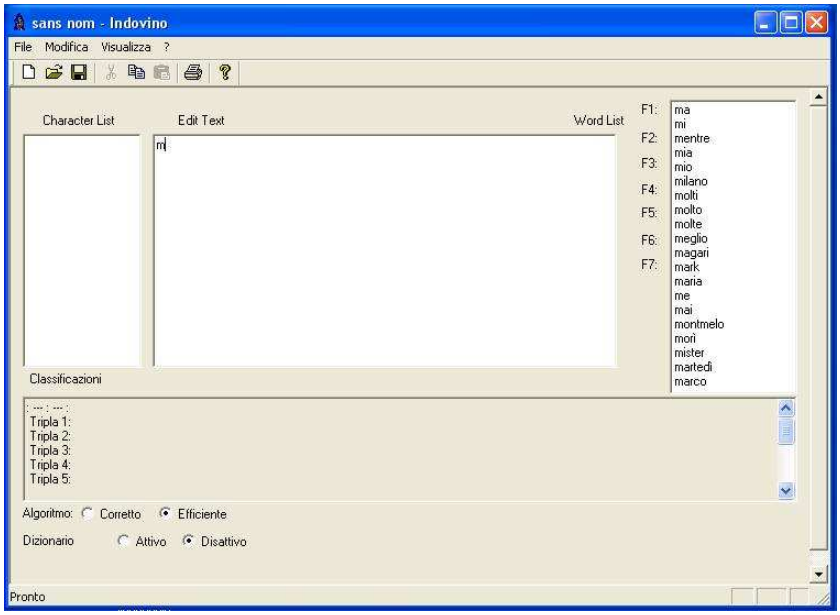

Figura 4.5: L'interfaccia Indovino in funzione dopo l'inserimento del primo carattere "M".

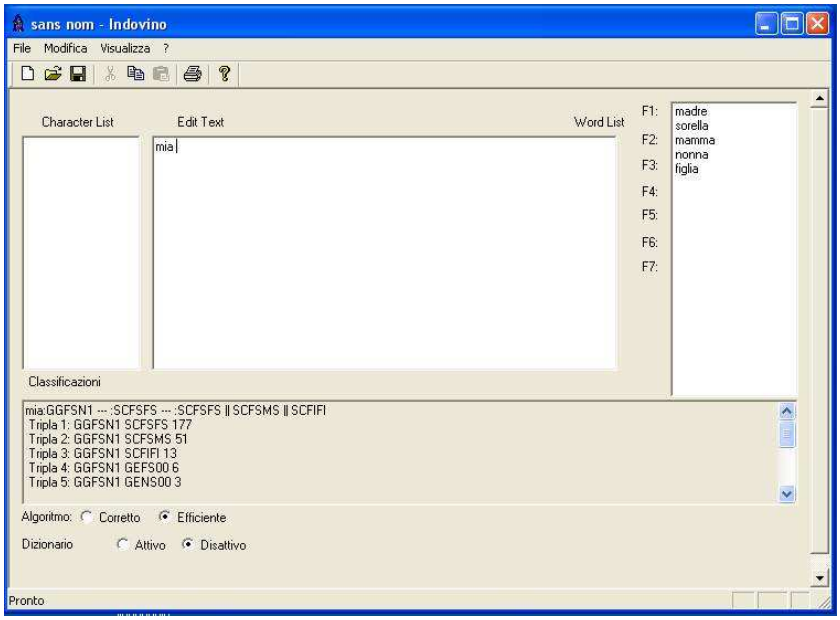

Figura 4.6: Selezione del suggerimento "mia" dalla Word List.

Notiamo che il predittore inserisce automaticamente uno spazio dopo la scelta del suggerimento, facendo risparmiare una digitazione. A questo punto il predittore classifica la prima parola inserita come Aggettivo Possessivo Femminile Singolare (GGFSN1) ed effettua la ricerca delle tre categorie grammaticali più probabili per la parola successiva, provenienti dai POS bigram. Le parole suggerite, come mostrato in Figura 4.6 appartengono alle classificazioni trovate, ovvero sostantivi femminili singolari, tali parole sono state trovate tra i word bigram di inizio frase. Come si può vedere dalla Figura 4.6, la parola desiderata "madre" si trova in prima posizione della lista dei suggerimenti.

Quando l'utente seleziona questo suggerimento, come mostrato in Figura 4.7, il predittore classifica la seconda parola come Sostantivo Comune Femminile Singolare (SCFS) e cerca le classificazioni più probabili per la terza parola, avendo a questo punto della predizione le due POS precedenti la ricerca di queste classificazioni avviene nei POS trigram (triple). Quindi prima cerca nei word bigram le coppie di parole che iniziano con la parola "madre". A questo punto avviene la combinazione lineare che ordina i suggerimenti trovati nella lista.

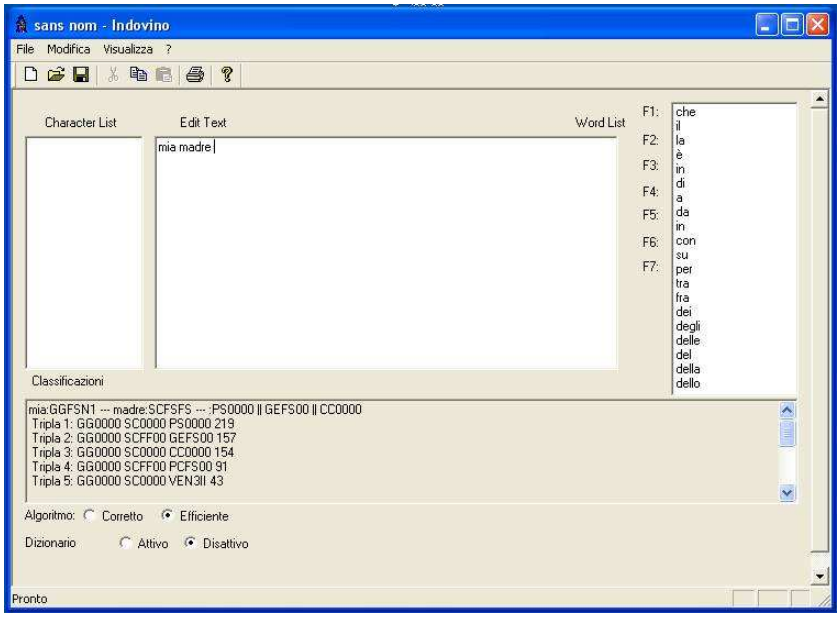

Figura 4.7: Selezione del suggerimento “madre”.

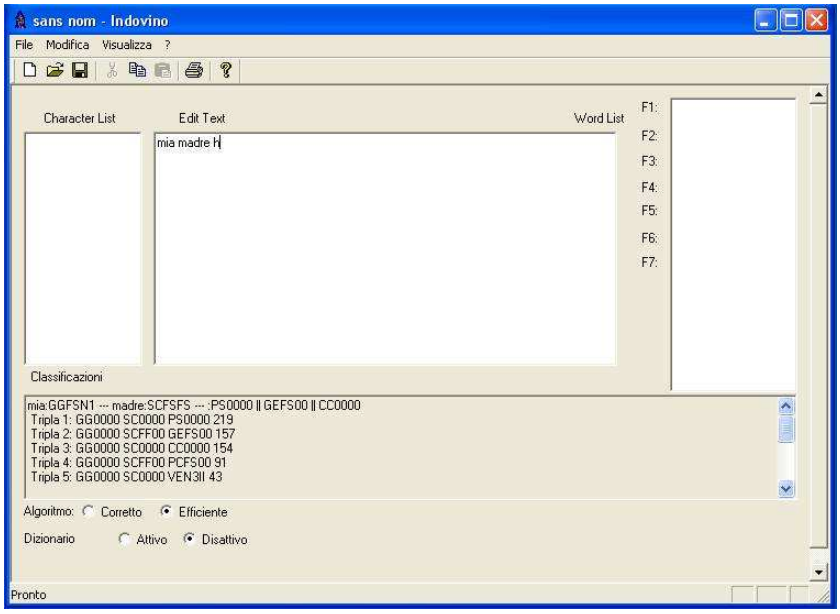

Figura 4.8: Inserimento del primo carattere “H” della terza parola.

Dato che la parola “ha”, la parola successiva che l’utente vorrebbe scrivere, non si trova tra i suggerimenti , come mostrato in Figura 4.8, è necessario inserire un altro carattere, dopo aver inserito il carattere “h”, la parola desiderata non si trova ancora tra i suggerimenti e quindi è stato necessario inserirla completamente.

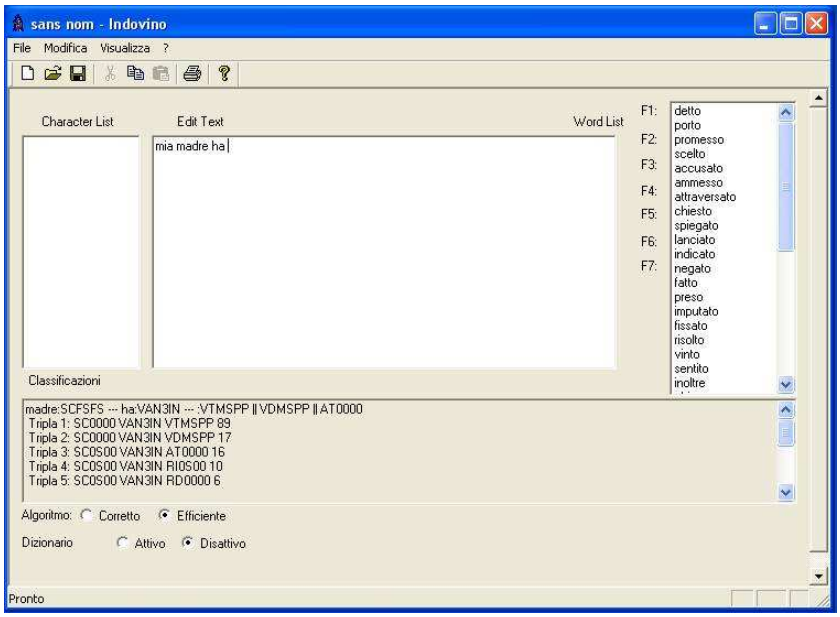

Figura 4.9: Inserimento della terza parola “ha”.

A questo punto, il predittore classifica la terza parola, “ha” (Figura 4.9), e viene effettuata la ricerca delle POS più probabili per la terza parola classificata come VAN3IN, ovvero, Verbo Ausiliare Avere 3° persona. La POS più probabile per la parola successiva è un verbo transitivo maschile singolare (VTMSPP), infatti quasi tutti i suggerimenti sono dei verbi transitivi. Il suggerimento atteso però non si trova nella lista quindi è necessario inserire il primo carattere della parola “preparato”, come mostrato in Figura 4.10.

Dopo l'inserimento del carattere "p" la parola desiderata si trova in nona posizione.

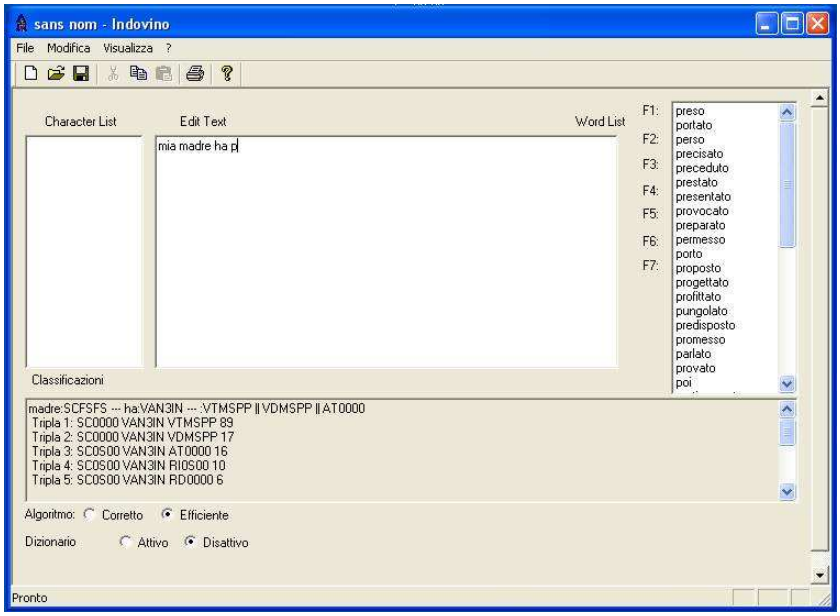

Figura 4.10: Inserimento primo carattere "p" della quarta parola.

Una volta inserito il suggerimento "preparato" i POS trigram vengono aggiornati, ovvero POS1 diventa POS2 e POS2 diventa POS3. Non avendo inserito nessun carattere successivo, il predittore propone le parole più frequenti (Figura 4.11). La parola desiderata "una" si trova tra i suggerimenti. A questo punto l'utente seleziona il suggerimento, Figura 4.12.

Nelle successive figure: Figura 4.13, Figura 4.14, Figura 4.15, Figura 4.16 viene mostrato il comportamento del predittore per la digitazione del resto della frase.

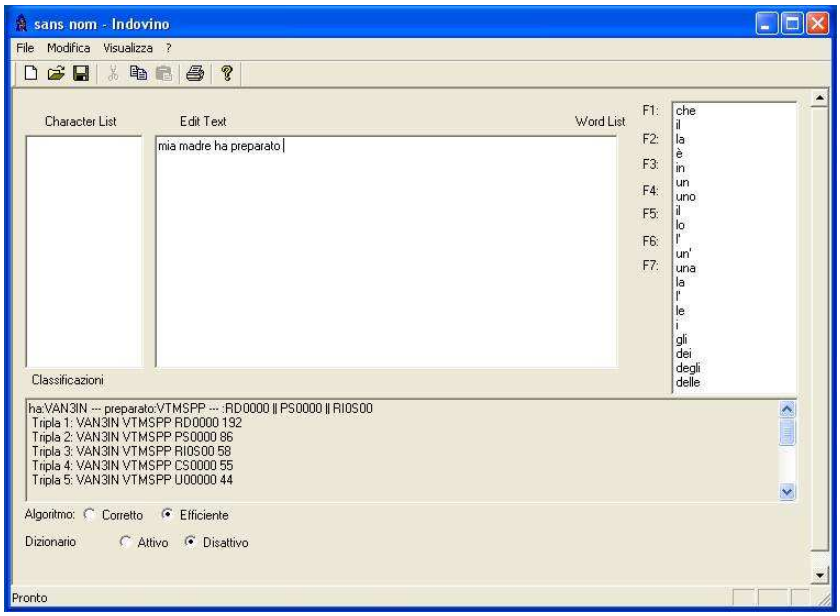

Figura 4.11: Selezione del suggerimento “preparato”.

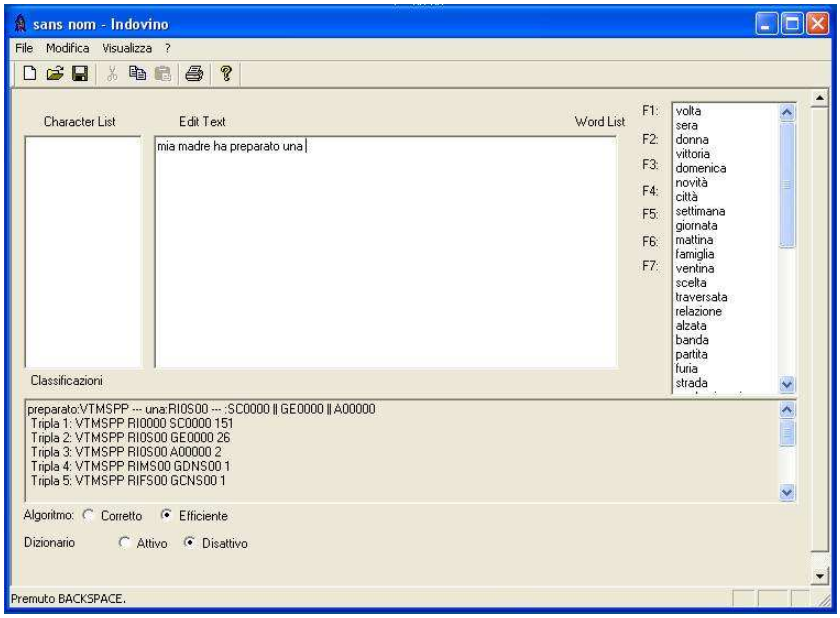

Figura 4.12: Selezione del suggerimento “una”.

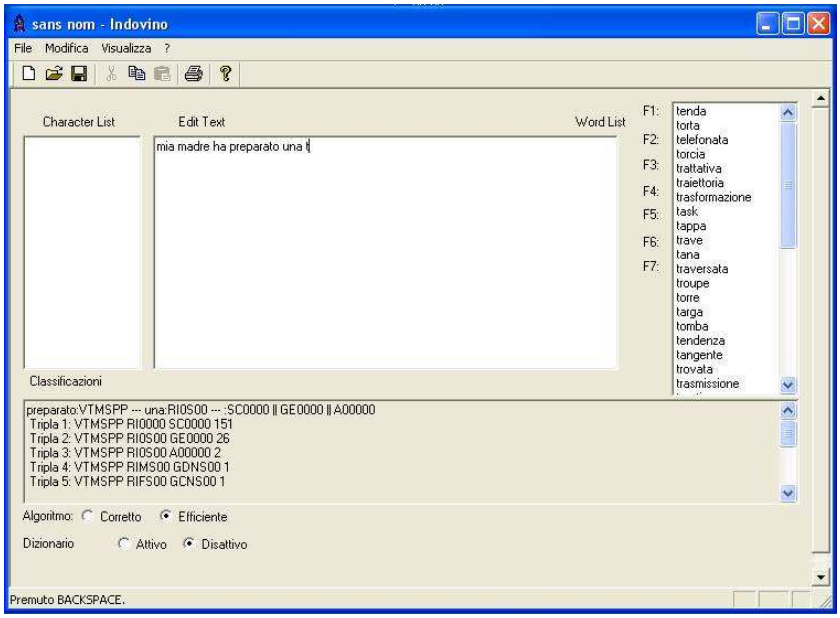

Figura 4.13: Inserimento del carattere “t”.

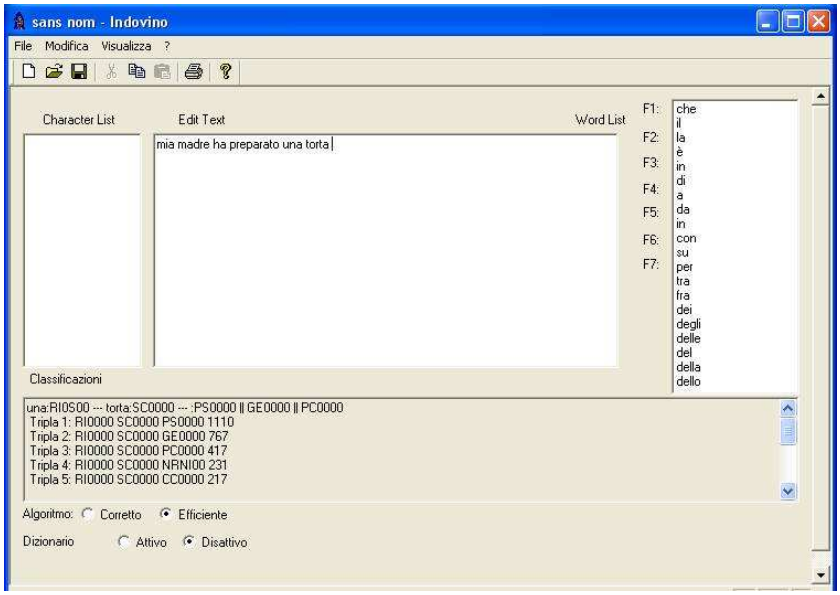

Figura 4.14: Selezione del suggerimento “torta”.

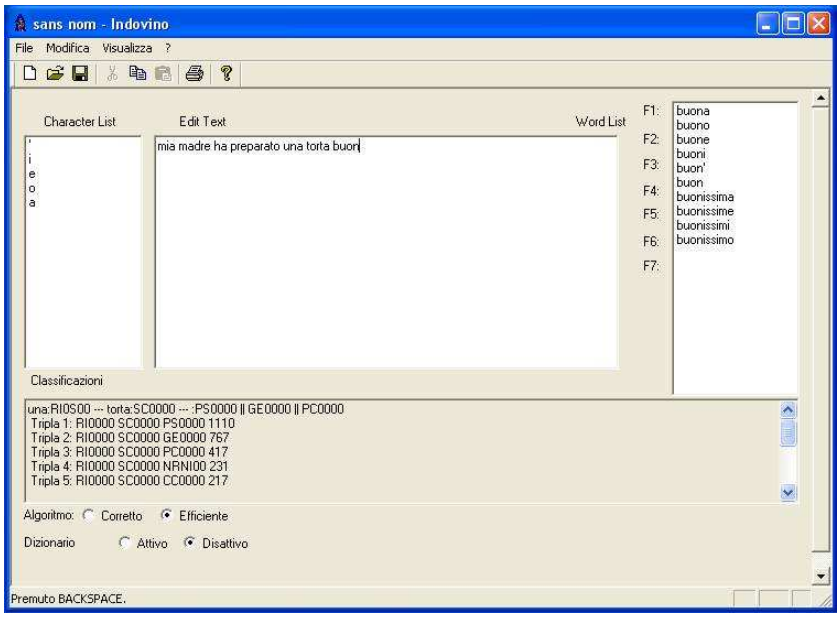

Figura 4.15: Inserzione dei caratteri “b”,“u”,“o” e “n”.

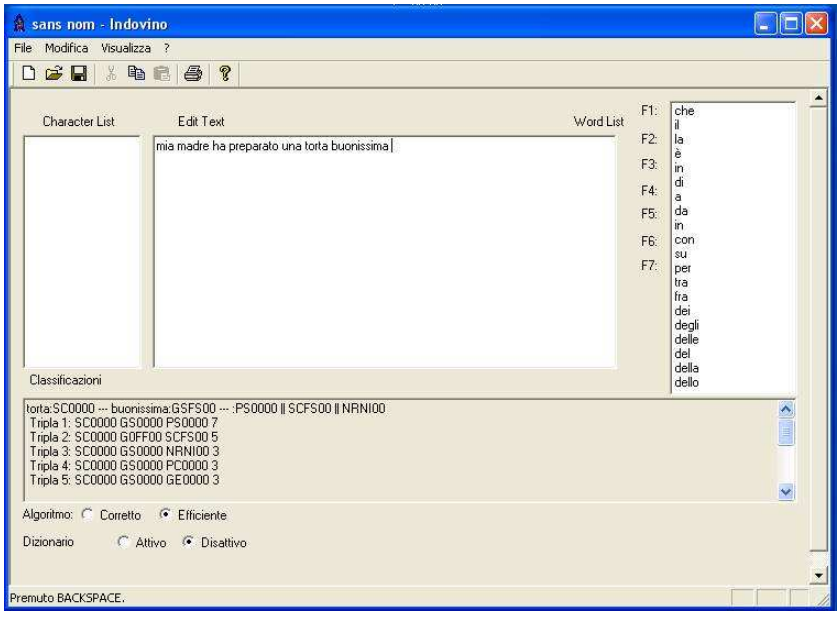

Figura 4.16: Selezione del suggerimento “buonissima” (fine frase).

Capitolo 5

Test e Verifiche

5.1 Metodologia di Test

In questo progetto di tesi avevamo come obiettivo principale lo sviluppo di un sistema di supporto capace di *velocizzare* l'attività di scrittura al computer e *minimizzare il numero di digitazioni*. Durante la fase di verifica sono stati effettuati diversi esperimenti per stimare la velocità di scrittura e il numero di digitazioni necessarie all'inserimento di un dato testo al fine di valutare l'efficacia della predizione.

Per valutare il sistema abbiamo introdotto una serie di parametri numerici che descriveremo in questa Sezione. Alcuni di questi parametri sono normalmente utilizzati nella valutazione di sistemi di scrittura assistita, mentre altri sono stati introdotti a seguito di considerazioni derivate dall'uso del predittore FastType.

5.1.1 Parametri

I parametri utilizzati e descritti di seguito sono cinque: il Keystroke Saving, il Word Type Saving, il Word 3-Ranking, il Keystrokes Until Prediction e la Copertura del Lessico.

- Il ***Keystroke Saving*** (**KS**) è la percentuale effettiva di caratteri risparmiati, ovvero che l'utente non ha dovuto digitare e corrisponde alla differenza, tradotta in percentuale, fra il numero di digitazioni necessarie per scrivere un brano di testo senza utilizzare alcun sistema di scrittura assistita e il numero di digitazioni necessarie per scrivere lo stesso brano utilizzando un predittore di parola.

  Indicando con K_{total} il numero totale di caratteri presenti nel testo, con K_{typed} il numero di caratteri digitati, si ha:

$$KS = \frac{K_{total} - K_{typed}}{K_{total}} \cdot 100$$

  Ad esempio, per digitare il brano di testo "repubblica" sono necessarie 10 digitazioni. Se dopo aver digitato "rep" (3 keystroke) il predittore suggerisce la parola "repubblica", l'utente può accettare il suggerimento (premendo un tasto, che influisce sul calcolo del keystroke saving) e lasciare che il predittore completi la parola. Poiché l'utente ha premuto 4 tasti (3 per scrivere "rep" e uno per accettare il suggerimento) invece dei 10 previsti, il KS è del 70%.

Anche gli spazi, infatti influiscono sul KS: il predittore FastType inserisce automaticamente uno spazio dopo aver completato una parola, risparmiando un ulteriore keystroke all'utente. Una situazione particolare è rappresentata dagli spazi prima della punteggiatura: poiché prima dei segni di punteggiatura non si inserisce uno spazio, se dopo aver inserito un suggerimento (e il relativo spazio), l'utente digita un segno di punteggiatura, il predittore cancella lo spazio che precede il segno appena digitato per garantire un'ortografia corretta.

- Il ***Word Type Saving*** (**WTS**) è la percentuale di tempo risparmiato dall'utente quando si avvale del completamento automatico delle parole rispetto a quando non dispone di un sistema di scrittura assistita come un predittore di parola. Matematicamente corrisponde alla differenza tra il tempo necessario a scrivere un brano di testo senza utilizzare alcun sistema di scrittura assistita diviso il numero di parole e il tempo necessario a scrivere lo stesso brano utilizzando un predittore di parola diviso il numero di parole.

  Indicando con T_n il tempo impiegato a digitare i testi di prova senza scrittura assistita e con T_a il tempo impiegato a digitare gli stessi testi ma approfittando del completamento automatico delle parole, si ha:

$$WTS = \frac{T_n - T_a}{T_n} \cdot 100$$

Ad esempio, se per digitare un brano di testo di 30 parole, senza sistemi di scrittura assistita, l'utente impiega 50 secondi e per digitare lo stesso brano con il predittore FastType l'utente impiega 30 secondi, il Word Type Saving è del 40%.

Maggiore è il WTS migliore sarà l'algoritmo di predizione.

E' chiaro che questo parametro dipende dal particolare utente e non può, in generale, costituire una misura "assoluta" come nel caso del Keystroke Saving. Nel paragrafo 5.1.2 descriveremo l'uso di un "utente" virtuale per la misura di questo parametro.

- Il ***Word 3-Ranking*** (**W3-R**) è la posizione della predizione corretta (ovvero della parola che stò scrivendo e che vorrei venisse completata automaticamente) nella lista di suggerimenti quando mancano 3 caratteri al suo completamento. Questo criterio è pertanto valido solo per le parole lunghe almeno 4 caratteri.

  Ad esempio, se desidero scrivere "repubblica" e dopo aver scritto "repubbl" la parola "repubblica" compare al 2° posto nella lista di suggerimenti, il word 3-ranking sarà 2.

  Indicando con $p_1 \dots p_n$ le posizioni dei completamenti corretti per n parole, si ha:

$$W3 - R = \frac{p_1 + p_2 + \ldots + p_n}{n}$$

ovvero la media delle posizioni.

- Il ***Keystrokes Until Prediction*** (**KUP**) è il numero medio caratteri premuti per visualizzare la parola desiderata nella lista di predizione. Viene calcolato come:

$$KUP = \frac{\sum_{w_i \in T} K_{typed}(w_i)}{Count(T)}$$

dove,

T è il testo da predire;

$K_{typed}(w_i)$ è il numero di caratteri digitati per completare la parola w_i prima che comparisse nella lista di predizione;

$Count(T)$ è il numero totale delle parole nel testo.

Ad esempio, se per la prima parola ho digitato due caratteri, per la seconda tre caratteri e per la terza quattro caratteri il numeratore $\sum_{w_i \in T} K_{typed}(w_i)$ varrà $2 + 3 + 4 = 9$ e $Count(T) = 3$ parole. Quindi KUP $= \frac{9}{3} = 3$.

Minore è il KUP migliore sarà l'algoritmo di predizione.

- La ***Copertura del lessico*** misura la capacità del dizionario di classificare le parole digitate dall'utente. Classificare correttamente le parole è necessario per la predizione basata sulla sintassi. Poiché questo cri-

terio dipende dal dizionario utilizzato e non dalla lunghezza della lista, il suo valore è costante.

Indicando con P_t il numero complessivo di parole presenti nel testo e con P_s il numero di parole sconosciute, si ha:

$$Copertura\ del\ lessico = \frac{P_s - P_t}{P_s} \cdot 100$$

Il parametro che appare più significativo è senza dubbio il Keystroke Saving, seguito, in ordine di importanza, dal Word Type Saving, soprattutto nel caso dell'uso del predittore da parte di persone disabili. Il Word 3-Ranking misura la qualità della predizione del sistema.

I restanti parametri risultano significativi per valutare e ottimizzare l'usabilità dello strumento.

5.1.2 Procedura di Test

Per effettuare i test necessari a valutare il predittore, è stata utilizzata l'interfaccia Indovino, con lo scopo di provare il componente di predizione delle parole e dei caratteri successivi.

L'interfaccia (come mostrato in Figura 5.1) è divisa in quattro parti: nella colonna a sinistra sono mostrati i caratteri predetti, al centro si trova la zona di composizione del documento di testo e nella colonna a destra sono visualizzati i suggerimenti per il completamento. Cliccando con il tas-

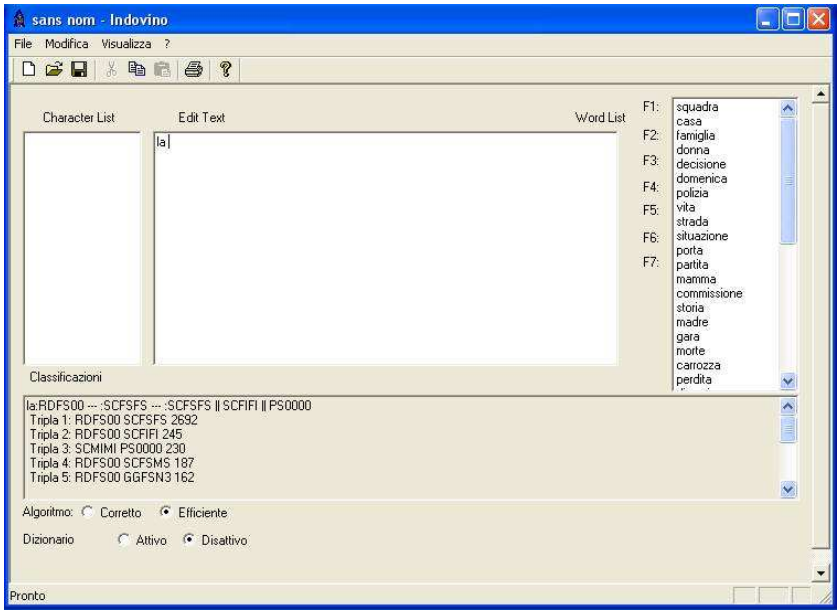

Figura 5.1: L'interfaccia Indovino.

to sinistro del mouse su un suggerimento della colonna a destra, Indovino lo inserisce nella zona di composizione, completando la parola che l'utente sta digitando, mentre cliccare su uno dei caratteri della colonna a sinistra è equivalente a premere il tasto corrispondente sulla tastiera. In basso sono visualizzate alcune informazioni sullo stato interno della libreria. Al termine della digitazione di un testo, Indovino produce automaticamente un file di testo che contiene i dati necessari a calcolare le misure di valutazione: numero di caratteri digitati, numero di caratteri totali, tempo impiegato a digitare il testo, posizione delle parole nella lista di suggerimenti. I dati vengono poi analizzati per ottenere i risultati definitivi.

Per immettere i testi in Indovino è stato realizzato un altro programma, denominato *"Rob Bottin"*, che automaticamente digita il testo e seleziona i

suggerimenti presentati nella colonna destra di Indovino per completare le parole. Questo "**utente simulato**" garantisce l'immissione dei caratteri a velocità costante, condizione essenziale per la misurazione del tempo medio di composizione delle parole. Per svolgere il suo compito, *Rob Bottin* utilizza le funzioni a basso livello di Windows per generare eventi di tastiera corrispondenti alla pressione dei tasti e ai clic del mouse.

In generale, "*l'utente virtuale*" o "*simulato*" è un programma che legge, in ogni testo usato per i test, una lettera alla volta. Dopo aver letto tutte le lettere, determina qual'è la predizione giusta per la posizione corretta o quella successiva. A questo punto viene chiamato l'algoritmo di predizione che ritorna la lista dei suggerimenti. L'utente simulato cerca la parola corretta nella lista dei suggerimenti, clicca il suggerimento e l'algoritmo di predizione raccoglie i dati necessari per la valutazione. L'utente virtuale è considerato come un utente perfetto: infatti nel caso in cui la parola si trovi nella lista dei suggerimenti l'utente simulato non la perde. Questo non è sempre vero per l'utente umano. L'utente può non vedere il suggerimento anche se si trova nella lista e ciò dipende da tanti fattori (per esempio dal numero di suggerimenti nella lista e dal tipo e grado di disabilità dell'utente).

I brani di testo immessi in Indovino per le valutazioni sono tratti da giornali e riviste, ma sono stati selezionati perché includessero diversi stili di scrittura: uno stile sintetico da giornalista, uno stile narrativo e ricco di dettagli, uno stile colloquiale e uno epistolare orientati alla comunicazione

tra due persone più che alla divulgazione. Queste fonti iniziali, provenienti da 40 testi, sono state raccolte in 4 file di testo:

File numero	Parole	Caratteri (spazi esclusi)	Caratteri (spazi inclusi)
1	711	3740	4451
2	541	2760	3301
3	266	1546	1811
4	577	2909	3485

Tabella 5.1: Testi test.

Da notare che per quanto sia parte integrante del sistema, il dizionario personale dotato di algoritmo di autoapprendimento non viene utilizzato in fase di test, per mantenere l'omogeneità dei risultati anche in caso di ripetizione dei test; se il dizionario personale fosse attivato il predittore assimilerebbe il lessico dei quattro file di testo descritti sopra e in caso di successive esecuzioni dei test, otterrebbe risultati molto migliori.

Il test sopra descritto è stato ripetuto quattro volte, provando a cambiare la lunghezza L della lista di suggerimenti. I valori provati per L sono 5, 10 e 20. Nelle prime tre esecuzioni, l'utente simulato aveva il compito di digitare i testi di prova usando sempre il completamento automatico delle parole, cioè selezionando il completamento giusto non appena questo compariva nella lista di predizione. In questi primi tre test sono stati misurati il Keystroke Saving, il Word Type Saving ed il Keystrokes Until Prediction. Nella quarta esecuzione del test, l'utente simulato aveva il compito di digitare ogni parola dei testi di prova fino al quart'ultimo carattere e solo allora cercare

nella lista dei suggerimenti (lunga 20 parole) un completamento automatico. Quest'ultimo test ha permesso di misurare il Word 3-Ranking. È stata infine effettuata un'ultima digitazione automatica, impostando l'utente simulato perché non utilizzasse il completamento automatico ma digitasse per intero i testi, così da avere il tempo "di riferimento" necessario per il calcolo del Word Type Saving, da confrontare con il tempo, registrato nelle prime tre esecuzioni del test, impiegato dall'utente simulato per completare la battitura dei testi avvalendosi del completamento automatico.

Un'altra tipologia di test ha riguardato il corpus specifico medico-radiologico. La valutazione è stata effettuata sottoponendo alcuni referti medici al predittore ed usando le risorse linguistiche radiologiche decritte nella Sezione 4.6.3 calcolando in questo caso solo il Keystroke Saving.

5.1.3 Risultati dei Test

Come descritto nella Sezione precedente il KS, il WTS e il KUP sono stati misurati tre volte, sperimentando con tre diverse lunghezze L della lista di predizione.

L	KS	WTS	KUP
5	41,15%	21,12%	2,85
10	45,26%	24%	2,67
20	47,9%	24,35%	2,48

Tabella 5.2: Risultati dei test.

Come si può osservare, quando la lunghezza della lista di predizione scende sotto 10 i parametri misurati peggiorano più velocemente: la differenza tra il KS con L = 5 e quello con L = 10 è all'incirca il doppio di quella presente tra L = 10 e L = 20 anche se la lista è stata ridotta solo di 5 unità e non di 10; una osservazione simile vale per il KUP, dato che la differenza è la stessa, ma anche in questo caso la lista è stata ridotta solo di 5; il WTS, diminuito dello 0,35% da L = 20 a L = 10, peggiora di quasi il 4% passando da L = 10 a L = 5. La scelta migliore per stabilire la lunghezza della lista di predizione è aggiungere un certo grado di personalizzazione all'interfaccia, lasciando scegliere all'utente la lunghezza della lista, mantenendola però entro 10, così che, dovendo mostrare una lista di lunghezza limitata, l'interfaccia possa avere caratteri grandi e ben visibili.

Il W3-R ha invece lo scopo di misurare la qualità dell'algoritmo di predizione, e dato che prescinde dalla lunghezza della lista di predizione, è stato misurato una sola volta, ottenendo:

$$W3 - R = 3,4 \; nel \; 58,5\% \; delle \; parole$$

Il ranking delle parole è pertanto sufficiente (la parola è in media entro le prime 5), ma troppo spesso l'utente non riesce a trovare la parola desiderata nella lista quando mancano 3 caratteri al suo completamento (nel 41,5% dei casi deve digitare la parola per intero).

Nella Figura 5.2 e Figura 5.3 sono riportati due esempi di test effettuati estratti dai file di testo citati precedentemente. In colore sono stati evidenziati i caratteri risparmiati, ovvero quelli completati dal predittore. Nel primo esempio, i caratteri totali sono 349 e i caratteri risparmiati sono 175, quindi il KS $\sim$ 50%. Nel secondo esempio, i caratteri totali sono 172 e i caratteri risparmiati sono 94, quindi il KS $\sim$ 45%.

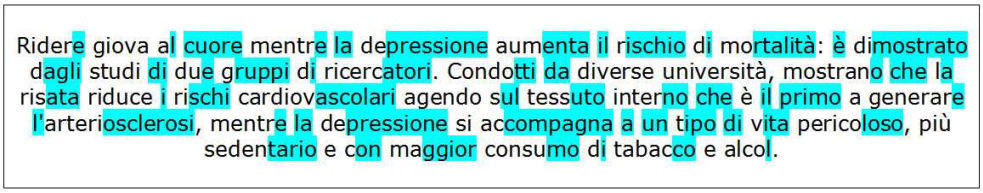

Figura 5.2: Esempio di test.

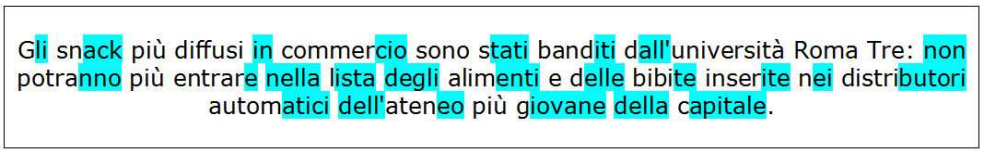

Figura 5.3: Altro test effettuato.

Determinazione dei Valori di α, β e γ

Per trovare i valori di α, β e γ siamo partiti da valori iniziali dei tre coefficienti compresi tra 0 e 1 e, tramite l'utente simulato, abbiamo effettuato dei test calcolando il KS. Variando ogni coefficiente con un passo di 0.01, siamo riusciti a calcolare la combinazione che restituisce il valore più alto di KS.

La combinazione ottimale dei tre valori che è stata trovata è la seguente: $\alpha = 0.48$, $\beta = 0.32$ e $\gamma = 0.2$.

Nel caso in cui il dizionario personale sia disattivato, i coefficienti della combinazione lineare sono due: α e β. Anche in questo caso siamo partiti da valori iniziali dei due coefficienti compresi tra 0 e 1 e, tramite l'utente simulato, abbiamo effettuato dei test calcolando il KS. Variando ogni coefficiente con un passo di 0.1, abbiamo calcolato la combinazione che restituisce il valore più alto di KS. La combinazione ottimale dei due valori che è stata trovata è la seguente: $\alpha = 0.6$ e $\beta = 0.4$.

Risultati con le Risorse Medico-Radiologiche

Per quanto riguarda i test effettuati sottoponendo diversi referti medici al predittore, e come sottolineato precedentemente, usando risorse linguistiche specifiche dell'ambito medico-radiologico, la percentuale di Keystroke Saving ottenuta è del 75%. Tale risultato è ottimo considerando che le triple usate per predire la POS, non sono state addestrate a partire da un corpus medico-radiologico.

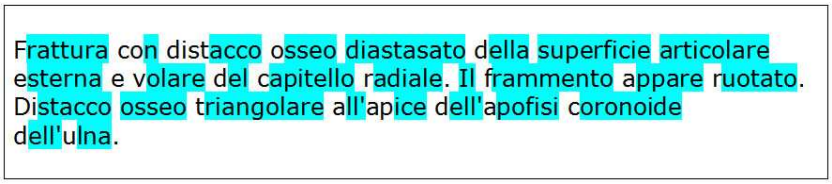

Figura 5.4: Test effettuato con le risorse medico-radiologiche.

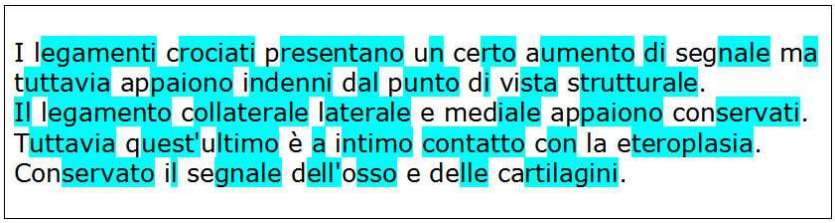

Figura 5.5: Altro test effettuato con le risorse medico-radiologiche.

Nella Figura 5.4 e Figura 5.5 sono presentati due esempi di test effettuati utilizzando i word bigram taggati radiologici ed il dizionario medico-radiologico. Come è possibile notare, i caratteri evidenziati sono molti. Tale risultato è riconducibile, in parte, alla grande quantità di risorse linguistiche a disposizione, in particolare i word bigram e, dall'altra parte, al fatto che la terminologia usata per la scrittura dei referti appartiene ad un lessico molto specializzato e quindi limitato.

Capitolo 6

Conclusioni e Sviluppi Futuri

6.1 Conclusioni

In questo lavoro di tesi è stata realizzata l'estensione di un sistema di predizione di parola con lo scopo di migliorarne la predizione e di accelerare il processo di scrittura per l'utente.

Il lavoro svolto è consistito nella creazione di nuove risorse linguistiche, da una parte, e nell'implementazione di un algoritmo capace di effettuare predizione efficace di parola, dall'altra.

Il sistema realizzato è capace di generare un dizionario personale costituito dalle parole digitate dall'utente, usate più spesso.

Ognuna delle risorse linguistiche create, ovvero triple di Part-of-Speech, Part-of-Speech bigram, word bigram taggati e dizionario personale determinano un meccanismo di predizione.

Il modello sintattico e statistico proposto in questa tesi combina tutti questi meccanismi di predizione, per ottenere una migliore predizione di parola.

I diversi test effettuati per verificare e convalidare il predittore di parola hanno dato risultati soddisfacenti. Il sistema realizzato permette di risparmiare in media il 48% di digitazioni (Keystroke).

Il sistema è stato testato anche su referti radiologici ed i risultati in questo contesto sono molto soddisfacenti, con un risparmio di digitazioni medio del 75%.

Il predittore di parola è stato realizzato principalmente per essere usato da persone disabili al fine di facilitare ed aumentare la velocità di inserimento del testo, tuttavia può essere integrato in tecnologie che presentano l'inserimento di testo rallentato come nei palmari o nei telefoni cellulari.

6.2 Sviluppi Futuri

Futuri sviluppi di questo lavoro possono essere orientati verso il miglioramento e l'estensione delle funzionalità e dell'utilità di questo sistema di predizione. Un possibile miglioramento verrebbe dalla raccolta di un più ampio e significativo insieme di testi da utilizzare per la costruzione delle risorse necessarie alla predizione statistica, in particolare ampliando il numero di coppie di parole (word bigram).

Un ulteriore miglioramento potrebbe derivare dalla realizzazione di un meccanismo di gestione ed espansione delle abbreviazioni.

Inoltre, sarebbe interessante prevedere altri dizionari di stile preferenziali in un determinato contesto di applicazione (per esempio tecnico, scientifico, giuridico).

Ulteriori sviluppi potrebbero derivare dall'integrazione di questo strumento in dispositivi portatili come palmari o telefoni cellulari, affiancando alla predizione una tecnologia vocale per la lettura dei suggerimenti durante la fase di scrittura, ottenendo così un ausilio fondamentale per i non vedenti.

Bibliografia

[1] A.S.P.H.I. Avviamento e Sviluppo di Progetti per ridurre l'Handicap mediante l'Informatica. `http://www.asphi.it/`.

[2] I.S.A.A.C. International Society of Augmentative and Alternative Communication. `http://www.isaac-online.org/`.

[3] World Health Organization. International Classification of Functioning, Disability and Health. `http://www.who.int/classifications/icf/`.

[4] World Health Organization Website. `http://www.who.int/en/`.

[5] C. Aliprandi, D. Barsocchi, F. Fanciulli, P. Mancarella, D. Pupillo, R. Raffaelli, and C. Scudellari. AWE, an Innovative Writing Prediction Environment. In *Proceedings of the 10th International Conference on Human-Computer Interaction*, pages 237–238, Crete, Greece, 2003.

[6] C. Aliprandi, D. Barsocchi, P. Mancarella, D. Pupillo, and R. Raffaelli. Trattamento Automatico della Lingua e Disabilità. *Media 2000 n. 208*, XXI-6(13):103–106, 2003.

[7] P. Baldi. *Calcolo delle Probabilità e Statistica*. McGraw-Hill, Milano, 1998.

[8] M. Baroni. Materiali per il Corso di "Linguistica Computazionale", Università di Bologna. `http://sslmit.unibo.it/~baroni/`, 2006.

[9] D. Barsocchi. Disabilità, Informatica, Linguistica: un'Istanza del Trinomio. Tesi di Laurea in Informatica, Università di Pisa, 2002.

[10] S. Bickel, P. Haider, and T. Scheffer. Predicting Sentences using N-gram Language Models. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 193–200, Morristown, NJ, USA, 2005.

[11] C. Callison-Burch and M. Osborne. Statistical Natural Language Processing. In A. Farghaly, editor, *A Handbook for Language Engineers*. CSLI Publications, 2003.

[12] N. Calzolari and A. Lenci. Linguistica Computazionale. Strumenti e Risorse per il Trattamento Automatico della Lingua. *Mondo Digitale*, III(2):56–69, 2004.

[13] J. Carlberger. Design and Implementation of a Probabilistic Word Prediction Program. Master's Thesis in Computer Science, Royal Institute of Technology (KTH), Stockholm, Sweden, 1997.

[14] N. Carmignani. Progetto di un Sistema di Word Prediction per Persone Disabili basato su Part-of-Speech Tagging. Tesi di Laurea in Tecnologie Informatiche, Università di Pisa, 2005.

[15] J. Eng and J. Eisner. Radiology Report Entry With Automatic Phrase Completion Driven by Language Modeling. *Radiographics*, 24(5):1493–1501, 2004.

[16] Y. Even-Zohar and D. Roth. A Classification Approach to Word Prediction. In *Proceedings of the 1st North American Conference on Computational Linguistics (NAACL'00)*, pages 124–131, Seattle, Washington, US, May 2000.

[17] A. Fazly. The Use of Syntax in Word Completion Utilities. Master's Thesis in Computer Science, University of Toronto, Toronto, Canada, 2002.

[18] A. Fazly and G. Hirst. Testing the Efficacy of Part-Of-Speech Information in Word Completion. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Workshop on Language Modeling for Text Entry Methods, Budapest, Hungary, 2003.

[19] N. Garay-Vitoria and J. González-Abascal. Word Prediction for Inflected Languages. Application to Basque Language. In *Proceed-*

ings of the 2nd ACL Workshop on Natural Language Processing for Communication Aids, pages 29–36, Madrid, Spain, July 1997.

[20] N. Garay-Vitoria and J. González-Abascal. Text Prediction Systems: a Survey. *Universal Access in the Information Society*, 4(3):188–203, February 2006.

[21] P. Geutner. Introducing Linguistic Constraints into Statistical Language Modeling. In *Proceedings of the International Conference on Spoken Language*, pages 402–405, Philadelphia, PA, USA, 1996.

[22] M. Ghayoomi and S. Assi. Word Prediction in a Running Text: A Statistical Language Modeling for the Persian Language. In *Proceedings of the Australasian Language Technology Workshop*, pages 57–63, Sydney, Australia, 2005.

[23] E. Gustavii and E. Pettersson. A Swedish Grammar for Word Prediction. Master's Thesis in Computational Linguistics, Uppsala University, Uppsala, Finland, 2003.

[24] S. Hunnicutt and J. Carlberger. Improving Word Prediction using Markov Models and Heuristic Methods. *Augmentative & Alternative Communication*, 17(4):255–264, 2001.

[25] F. Jelinek. Self-Organized Language Modeling for Speech Recognition. In *Readings in Speech Recognition*, pages 450–506. San Francisco, CA,

USA, 1990.

[26] F. Jelinek and R. Mercer. Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, North-Holland, Amsterdam, 1980.

[27] D. Jurafsky and J. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[28] S. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401, 1987.

[29] J. Li. Modelling Semantic Knowledge for a Word Completion Task. Master's Thesis in Computer Science, University of Toronto, Toronto, Canada, 2006.

[30] J. Li and G. Hirst. Semantic Knowledge in a Word Completion Task. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 121–128, Baltimore, MD, USA, 2005.

[31] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, May 1999.

[32] S. Mohammad and T. Pedersen. Complementarity of Lexical and Simple Syntactic Features: The SyntaLex Approach to Senseval-3. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 159–162, Barcelona, Spain, 2004.

[33] S. Palazuelos-Cagigas, S. Aguilera-Navarro, J. Rodrigo-Mateos, J. Godino Llorente, and J. Martín-Sánchez. *Considerations On The Automatic Evaluation Of Word Prediction System*, volume Augmentative and Alternative Communication: New Directions in Research and Practice, pages 92–104. Whurr Publishers, In F. Loncke, J. Clibbens, H. Arvidson and L. Lloyd edition, 1999.

[34] R. Raffaelli. Un Ambiente per lo Sviluppo di Grammatiche basato su un Parser Inverso, Parallelo e Seriale. IBM Pisa, 1992.

[35] R. Rosenfeld. Two Decades of Statistical Language Modeling: Where Do We Go From Here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000.

[36] N. Ruimy. Il modello Lessicale SIMPLE: dal Monolingue al Bilingue. In *Tercero Seminario de la Escuela Interlatina de Altos Estudios en Lingüística Aplicada*, San Millán de la Cogolla, Spagna, 2003.

[37] N. Ruimy, M. Monachini M, and N. Calzolari. Un Lexique Électronique Multi-Niveaux de l'Italien. In *Proceedings of XVII International Congress of Linguists Prague*, Prague, Czech Republic, 2003.

[38] S. Simula. Interfaccia Utente per un Sistema di Word Prediction. Tesi di Laurea Triennale in Informatica, Università di Pisa, 2006.

[39] Synthema. Lexical System Server, Application Program Interface Programming Reference. Pisa, 1995.

[40] M. Vescovi. Soothsayer: un Sistema Multi-Sorgente per la Predizione del Testo. Tesi di Laurea in Ingegneria Informatica, Politecnico di Milano, 2005.

[41] M. Wester. User Evaluation of a Word Prediction System. Master's Thesis in Computational Linguistics, Uppsala University, Uppsala, Finland, 2003.

Ringraziamenti

I miei ringraziamenti più sinceri vanno al Prof. Paolo Mancarella e al Dott. Carlo Aliprandi per avermi dato l'oppprtunità di svolgere questo lavoro di tesi, a Nicola e Michele per la costante e sincera collaborazione.

Un grazie particolare va a mio padre per il prezioso aiuto e per i suoi saggi consigli. Ringrazio inoltre la mia famiglia per avermi saputa incoraggiare nei momenti più difficili e per aver sempre avuto fiducia in me. A Enrico per essermi sempre stato vicino e per avermi sostenuta con amore, ad Anna, Francesco e Simona per la loro amicizia ed il loro supporto, ai miei compagni di studio: Francesca e Alessio e a tutti coloro che in questi anni hanno sempre creduto in me.